Wondermike Software
präsentiert



OSS DOS - XL

DISK - OPERATING - SYSTEM

Benutzerführung und Referenzhandbuch

DOS XL ist das neueste DOS in der Serie der Firma Optimized Systems Software.

DOS XL Version 2.3 ist ein direkter Nachfahre und völlig File kompatibel mit:

ATARI DOS 2.0
OS/A+ version 2.0
OS/A+ version 2.1
DOS XL version 2.2

Diese Ausgabe des DOS XL Handbuches ist, wie wir hoffen, zum einen ein benutzerfreundlicher Führer für noch nicht so fortgeschrittene User, die mehr die "normalen" Anwendungen bevorzugen und zum anderen ein wirkliches Referenz-Handbuch für das gesamte DOS XL - Diskettenbetriebssystem.

Welchen Teil des DOS XL - Handbuches Sie zuerst lesen, hängt von Ihrem Erfahrungsschatz und Ihren Ambitionen ab:

-- Sollten Sie nicht selbst programmieren, so brauchen Sie nicht mehr als Kapitel 1 bis Kapitel 3 zu lesen.

-- Wenn Sie in ATARI BASIC programmieren, sollten Sie vor dem Benutzen von DOS XL auf jeden Fall Kapitel 4 und Ihr BASIC - Handbuch gelesen haben.

-- Sind Sie ein Assembler - Programmierer, dann sollten Sie unbedingt das ganze Handbuch und speziell Kapitel 10 durcharbeiten.

-- Sollten Sie schließlich das DOS XL automatisieren wollen, was Ihnen erlaubt andere Dinge zu tun, während Ihr Computer arbeitet, dann ist das Kapitel 8 das Richtige für Sie.

Wenn Sie natürlich, unabhängig von Ihrer Erfahrung und Ihrem Anwendungsziel, die Befehlseinengung des DOS XL - Menüs langweilt, dann sollten Sie Ka~~pitel 5~~ und 6 berücksichtigen.

Was auch immer Sie mit dem DOS XL tun, wir hoffen, daß Ihnen dieses Handbuch helfen wird.

Inhaltsverzeichnis

Kapitel 1: Einführung

Dieses Handbuch ist eine Addition zu dem des OS/A+, das dem DOS XL vorangegangen ist. Ab und zu wird auf Teile dieses OS/A+ Handbuches für spezielle Informationen verwiesen.

1.1 Was man benötigt.

DOS XL braucht 32K Speicherplatz und arbeitet mit allen ATARI - Computern zusammen (außer ST - Serie). Obwohl DOS XL für alle User entworfen wurde, kann es zusätzlich durch die OSS SuperCartridges und zusätzlichen Speicherbereich der XL-Geräte erweitert werden.

1.2 Was ist ein DOS ?

Die Aufgabe des DOS XL ist eine Kommunikation zwischen Computer und Diskettenstation(en), Drucker und anderen Peripheriegeräten zu ermöglichen. DOS XL beinhaltet Kommandos und Hiflsprogramme, die es ermöglichen:

1. Informationen in Files auf der Diskette zu ordnen.
2. Diese Informationen einfach und präzise abrufen zu können.
3. Andere Anwendungprogramme, wie z.B. BASIC XL, MAC/65, BUG/65, ATARI BASIC etc. zu benutzen.
4. Die Kontrolle über den Computer entweder an das Betriebssystem, Cartridges oder an ein Programm auf der Diskette zu übergeben.

DOS XL ist das einzige für ATARI - Computer erhältliche DOS, das es ermöglicht zwischen einem einfachen, benutzerfreundlichen Auswahlmenü und einem schnelleren, ausgefeilten Kommando-Prozessor auszuwählen. Wenn Sie DOS XL zusammen mit der OSS Super Cartridges anwenden, stehen Ihnen noch zusätzlich 5K Speicher zur Verfügung.

1.3 Disk Files

Genauso wie ein Audio Tape eine Reihe von Liedern beinhalten kann, können Disketten eine bestimmte Anzahl von Files speichern (bis zu 64 pro Diskettenseite). Diese Files können Programme oder Daten in Text- oder irgendeiner anderen Form enthalten. Anders als bei Cassetten-Files muß jedes Disk-File einen Namen erhalten, unter dem man es jederzeit aufrufen kann.

Die Regeln für DOS XL - Filenamen sind:
- Ein bis acht Buchstaben Länge
- Optional gefolgt von einem Punkt und einem ein- bis dreistelligen Anhang(Extender)
- Es sind nur die Buchstaben A-Z und die Zahlen 0-9 erlaubt
- Das erste Zeichen muß ein Buchstabe von A-Z sein.

Der Teil des Filenamen, der vor dem Punkt steht nennt man den eigentlichen Programmnamen und das was nach dem Punkt kommt den Extender(= Anhang). Obwohl jeder die Extender natürlich beliebig benennen kann, haben sich folgende Bezeichnungen zur Unterscheidung der verschiedenen Aufzeichnungs-und Filetypen eingebürgert:

Extender:	Vorgeschlagene Verwendung:	Beispiel:
SAV	"SAVE"d BASIC File	GEORGE.SAV
LIS	"LIST"ed BASIC File	PROGR1.LIS
ACT	ACTION! Quellfile	MIKE.ACT
M65	"SAVE"d MAC/65 File	SORT1.M65
OBJ	binäres Objectfile	BILL.OBJ
COM	DOS XL Hilfsprogramm	COPY.COM
EXC	DOS XL Ausführungsfile	STARTUP.EXC
SYS	Systemprogramm, reserviert für DOS XL(und natürlich alle anderen DOS auch)	

In den meisten Fällen muß vor dem Programmnamen noch das Peripheriegerät, das angesprochen werden soll, angegeben werden, damit das DOS weiß, auf welchem Laufwerk es ein bestimmtes Programm zu suchen hat.

Das Format schaut so aus:

Dn: oder D:, wobei das "n" für die Nummer des anzusprechenden Laufwerkes steht.(zwischen 1 und 4, je nachdem wie viele Drives man besitzt).

Gibt man lediglich D: an, so setzt das System voraus, daß die Station 1 gemeint ist, was natürlich recht arbeitssparend ist, wenn man sowieso nur ein Laufwerk hat.

In einigen Fällen dürfen die Filenamen auch sog. "wild-cards" beinhalten, d.h. die Zeichen "?" und "*". Ein Fragezeichen ersetzt jeden Buchstaben innnerhalb eines Filenamens und das Sternchen darf für jede beliebige Anzahl von Buchstaben und/oder Zahlen verwendet werden;z.B.:AB*.C?? würde ABX.CCC,AB.CUR,ABCDEF.CNN etc.ansprechen. Wild-card-Kommandos dürfen in den folgenden DOS XL Menü- Kommandos verwendet werden:
Files on Disk(Disketteninhaltsverzeichnis), Copy files(Files kopieren), Erase files(Files löschen), Protect files(Files schützen) und Unprotect Files(Files entschützen)
Zu diesem Thema mehr in Kapitel 3. Weitere Informationen im Umgang mit Wildcards in Kapitel 5.

1.4 Andere Peripheriegeräte

Die ATARI Personal Computer akzeptieren alles außer den "Innereien" wie RAM,ROM und Prozessorchips als externe Peripherie, wobei sogar schon einige "externe" Geräte, wie die Tastatur und der Bildschirmeditor(E:) mit dem Computer geliefert werden. Einige der anderen Geräte sind Diskettenstation, Kassettenrecorder und Drucker. Wenn DOS XL nach einem Filenamen fragt, muß nicht unbedingt eine Diskettenstation als Datenträger genannt werden. Die anderen Peripherials werden durch einen Buchstaben(meist der Anfangsbuchstabe der englischen Bezeichnung für das entsprechende Gerät) und evtl. durch eine nachgestellte Zahl(bei mehreren Geräten,z.B.: D1:,D2:,D3:,etc.) benannt. Der sog. Device - Name muß von einem Doppelpunkt beendet werden. Im folgenden findet sich eine kleine Auflistung von Device-Namen, die im Zusammenhang mit einem unmodifizierten DOS XL verwendet werden dürfen:

C: Der Programm-Rekorder, der sowohl mit Ein-als auch mit Ausgabe arbeitet. Der Rekorder kann entweder als In-oder als Output-Gerät, niemals jedoch zu beidem gleichzeitig, verwendet werden.

D1: - D8: Diskettenstation(en), die ebenfalls sowohl mit Ein-als auch mit Ausgabe arbeiten, dies jedoch im Gegensatz zum Programmrecorder auch gleichzeitig. Bei diesen Geräten muß, wie schon beschrieben, ein Filename angegeben werden.

E: Der Bildschirm- oder Screen-Editor, ebenfalls mit Ein-und Ausgabe. Der Bildschirmeditor simuliert einen Text-Editor/Textverarbeitung, wobei dieser die Tastatur als Eingabe- und den Bildschirm als Ausgabegerät verwendet. Diesen Editor verwendet man bei der Eingabe von Zeichen im normalen BASIC(oder auch BASIC XL). Wenn man hier keinen extra Kanal angegeben hat, verwendet das BASIC den Kanal 0.

K: Keyboard(=Tastatur), verwendet nur Eingabe(logisch) und erlaubt Zugriff zur Tastatur ohne E:.

P: Parallel-Anschluß an der ATARI-850-Schnittstelle,verwendet nur Output. Normalerweise wird P: für einen Parallel geschalteten Drucker verwendet, so daß sich die Bezeichnung sowohl für Parallel Port als auch für Printer gleichermaßen eingebürgert hat.

R1: - R4: Die 4 RS232-Anschlüsse an der ATARI-850-Schnittstelle, die wiederum beides, Ein-und Ausgabe verwenden. Dieses Peripheriegerät ermöglicht es dem ATARI-Computer mit RS232-kompatiblen Geräten, wie Terminals, Plotter und Modems zu kommunizieren. Verwendet man R: ohne Nummer, so wird R1: angenommen.

S: Das Screen-Display(Bildschirmgestaltung) arbeitet auch mit Ein-und Ausgabe, wobei man hier jeden Buchstaben oder Graphikpunkt an beliebige Stellen auf dem Bildschirm mit Hilfe der Cursorposition abbilden kann.

1.5 DOS XL Kommando Betriebsarten

Ein ganz besonderer Vorteil des DOS XL, ist seine Möglichkeit, Kommandos in zwei verschiedenen Betriebsarten einzugeben, d.h. der User kann selbst auswählen, ob er im Menü-Modus oder im Kommando-Prozessor-Modus arbeiten will. Für jene, die noch nicht so vertraut sind mit Menügeführten Systemen, wie auch z.B. das ATARI-DOS: ein Menü ist ganz einfach eine Auflistung von Kommandos am Bildschirm, von denen man dann das gewünschte auswählen kann. Sollten weitere Informationen benötigt werden, so fragt das System diese vom User ab. Auf diese Weise muß man sich keine Kommandos merken, sondern hat sie jederzeit "griffbereit" vor sich stehen.

Die andere Möglichkeit dem DOS XL Befehle mitzuteilen, ist der Kommando-Prozessor-Modus, oder kurz "CP", bei dem man keine Befehlsliste erhält. Hier muß man die Kommandos auswendig kennen, was vielleicht auf den ersten Blick etwas umständlich erscheint, wenn man aber einmal eine bestimmte Praxis erlangt hat, läßt es sich hier viel schneller arbeiten. Weiterhin sind einige spezielle Vorzüge des DOS XL nur von hier aus zugänglich.

Zu Anfang, wenn man die DOS XL-Diskette auspackt, befindet man sich im Menü-Modus und erhält die Befehlsliste, wobei angenommen wird, daß man diesen Modus wirklich nur solange benutzt, bis man eine gewisse Erfahrung im Umgang mit DOS XL gesammelt hat. An dieser Stelle sei auf die Kapitel 4 und 5 verwiesen, die den Umgang mit dem CP detailliert erklären. Sollte man dann den Wunsch haben sofort nach Einladen des DOS XL im CP-Modus zu sein, so wäre zunächst das Kapitel 8 zu lesen, das Informationen enthält, wie man das DOS XL in dieser Richtung modifiziert.

Im folgenden werden einige Unterschiede zwischen dem Menü- und dem CP-Modus angegeben:

<u>Das DOS XL Menü</u> hat die folgenden Vor-und Nachteile:

Vorteile:
1. Man muß sich die Namen der DOS - Kommandos nicht merken und wird an den entsprechenden Stellen nach weiteren Informationen gefragt.
2. Diejenigen User, die schon mit dem ATARI DOS 2.0 oder 2.5 gearbeitet haben, werden sofort mit dem DOS XL vertraut sein.

Nachteile:
1. Bei der Verwendung des Menü-Modus werden ca.2K mehr Speicher benötigt, obwohl das nicht so sehr ins Gewicht fällt, wenn man die OSS SuperCartidge verwendet.

<u>Der DOS XL CP</u> (Kommando-Prozessor) hat die folgenden Vor-und Nachteile:

Vorteile:
1. Sobald man die ersten Befehle auswendig weiß, ist der CP-Modus in der Anwendung schneller als der Menü-Modus.
2. Diejenigen User, die bereits mit anderen Systemen, wie z.B. Apple DOS, CP/M, UNIX oder OS/A+ gearbeitet haben, werden den CP-Modus konventioneller und vertrauter finden.
3. Im Gegensatz zum Menü-Mode verbraucht der CP weniger Speicherplatz.

Nachteile:
1. Man wird nicht nach der jeweiligen Eingabe gefragt, die zum Gelingen der jeweiligen Operation notwendig ist und muß die verschiedenen fundamentale~~n Befe~~hle kennen, um die Vorzüge des CP voll ausnutzen zu können.

1.6. überblick über die DOS XL - Struktur.

DOS XL(und natürlich ATARI's Betriebssystem) verwendet ein Software-Konzept, welches in einem strukturierten und in bestimmter Weise aufgebauten Schema liegt. Im speziellen gesagt: Von Applikationsprogrammen wird erwartet, daß sie ihre Ein-und Ausgabe über die CIO (Zentrale Ein/Ausgabe-Routine) abwickeln;d.h.die CIO ist ein Verwalter, der die Anforderungen eines Applikationsprogrammes untersucht und die nötigen Untergeordneten Anforderungen an die Peripheriegeräte-Verwalter weitergibt.

Beim ATARI können diese Verwalter wiederum die SIO (serielle Ein/Ausgabe Routine) anrufen, die einen Kanal für die Kommunikation mit dem Peripheriegerät über den seriellen Bus herstellen soll(Selbstverständlich fallen hierunter nicht der Bildschirm(E:) und die Tastatur(K:), die keine Verbindung zum seriellen Bus haben.). Das Peripheriegerät schließlich erhält die SIO-Anfrage und führt den aktuellen IN/OUTPUT aus.

```
                                                      Menü
                             BASIC Program             !
Applikations                   des Users               !
programm                           !                   !
    !                          BASIC XL        Console Processor
    !                              !                   !
    ---------------------Eines von diesen-------------
                                   !
                              -----------
                              !   CIO   !
                              -----------
                                   !
    ----------------------------------------------------
    !             !                   !             !
  Disk          Drucker             Tastatur      Bildschirm
  Device        Device               Device       Device
  Driver        Driver               Driver       Driver
    !             !                    !             !
  --------------------           (ATARI Tastatur)  (TV/Monitor)
  !       SIO        !
  --------------------
     !              !
(Disk drive)    (Drucker)
```

Abbildung 1-1

Generell gesagt, gibt es keinen Grund irgendeinen oder mehrere Teile dieser hierarchischen Struktur nicht durch einen anderen, ebenbürtigen Teil zu ersetzen. Bei den ATARI-Computern wird auch das DOS XL(oder jedes andere DOS auch) nur an einen bestimmten Platz in dieser Anornung eingefügt und zwar nur, wenn beim Einschalten eine Diskettenstation angeschlossen ist. Einige Hersteller haben z.B. die Kontroll-Devices für einen Drucker oder den Bildschirmeditor geringfügig verändert und ersetzt.

Leider kann man aber auch nicht sagen, daß man jedes dieser Teile einfach so ohne jeden Nebeneffekt austauschen kann, ganz einfach weil eine ganze Menge an Software für den ATARI direkt auf die einzelnen Peripherials zugreift und die Struktur einfach umgeht. Diese Praktiken sind aber zum Glück nicht in der Mehrzahl anzutreffen, da sonst eine ordentlich funktionierende Computerstruktur nicht so ohne weiteres denkbar wäre. Auf jeden Fall sollte man wenigstens die gravierendsten Veränderungen, die in manchen Anwendungsprogrammen in Bezug auf den Peripheriegeräte-Zugriff gemacht wurden, im Auge behalten und sehen, welche Maßnahmen man treffen kann, um das System dahingehend abzuändern.

Generell sind die schlimmsten Programme solche, wie VISICALC oder MICROSOFT BASIC (Disk-Version), die Veränderungen in der Speicherbelegung vornehmen und bestimmte Diskettenformate benutzen. Diese Programme(und andere) beinhalten meist ein eigenes Betriebssystem, das sich ebenfalls auf der Diskette befindet. Obwohl diese, wie gesagt keine Verwendung für die Device-Verwalter haben, funktionieren sie dennoch in ihrem abgegrenzten inneren System mit der Diskettenstation, nur eben anders.

Wie Sie vielleicht bereits in Abbildung 1-1 bemerkt haben, sind der CP und das Menü keine unbedingten Teile des Systems. Der CP funktioniert als eine einfach zu bedienende Schnittstelle zwischen dem Menschen an der Tastatur und dem Maschinensprache-Level der CIO-Aufrufe und das Menü ermöglicht einen noch einfacheren Zugriff auf den CP und von hier aus zum Rest des Systems.

1.7 Fachausdrücke

Die folgenden Begriffe werden im weiteren Verlauf dieses Handbuches verwendet werden. Ihre Definitionen werden im folgenden aufgeführt, so daß man sich mit ihnen vor dem Weiterlesen schon einmal vertraut machen kann und dient ebenso als Nachschlagemöglichkeit im weiteren Verlauf der Beschreibungen.

Ausdruck:	Denificion:
DOS	Abkürzung für Disketten-Betriebssystem
boot,booting Boot-Programm	Der Einladvorgang von DOS XL oder jedem anderen
master diskette	Die DOS XL Master Diskette oder eine Arbeitskopie
bootable disk	Jede Diskette, die das File DOS.SYS enthält,mit der man z.B.das DOS XL in den Speicher laden kann.Um hiervon Gebrauch machen zu können, muß sich das File DOSXL.SYS ebenfalls auf dieser Diskette befinden.
Filename	Die Buchstabenfolge, mit der auf ein bestimmtes Programm auf der Diskette hingewiesen wird.
Filespec	Ein Filename, der die Wildcard-Zeichen(?,*) beinhalten darf.
prompt	Jede Mitteilung des DOS XL, die einem etwas erklärt oder einen auffordert eine Antwort einzugeben.
<RETURN>	Die RETURN-Taste ist zu drücken.
Utility(programm)	Ein Hilfsprogramm, das von DOS XL benutzt wird um bestimmte Diskettenoperationen auszuführen.

Kapitel 2: Mit DOS XL vertraut werden.

2.1 Das Booten der DOS XL Master Diskette.

Zuerst einmal ist abzuwägen, ob man die DOS XL - Version mit einfacher oder mit doppelter Dichte verwenden will, wobei es sich wahrscheinlich herausstellen wird, daß man für bestimmte Anwendungen im einen Fall die einfache und im anderen die doppelte Schreibdichte besser gebrauchen kann. Aus diesem Grunde ist die DOS XL Master Disk auch beidseitig beschrieben, auf der Vorderseite befindet sich (meistens) die Version mit der einfachen und auf der Rückseite die mit der doppelten Dichte. Aber warum gibt es in manchen Fällen für jede Version besondere Vorzüge? Im folgenden sind die wohl gravierendsten Gründe aufgeführt.

Einfache Schreibdichte: In erster Linie dient diese Version zur Kompatibilität mit den 810er Laufwerken. Die Single-density-Ausführung von DOS XL ist vollkommen file-kompatibel mit dem ATARI DOS 2.0,d.h. alle Operationen, die unter DOS 2.0 funktionieren, sollten auch mit DOS XL zusammen arbeiten.

Gibt es da auch Ausnahmen ? Ja ! Verschiedene Hersteller bieten selbstbootende Disketten an (d.h. man muß die Diskette lediglich in das Laufwerk legen und den Computer einschalten), die Aufrufe und Beziehungen nur zum ATARI DOS beinhalten. Es gibt keine Möglichkeit für das DOS XL mit diesen Programmdisketten zusammenzuarbeiten. Auf der anderen Seite ist das wohl doch kein allzugroßes Problem, da auf diesen selbstbootenden Disketten dann sicherlich ein ATARI DOS implementiert ist. Um solche Disketten zu verwenden ist nichts besonderes zu beachten, außer natürlich die Betriebsanleitung des Herstellers.

Doppelte Schreibdichte: Generell werden alle Programme, die mit Cartridge-basierten(oder in den Computer eingebauten) Sprachen erstellt wurden, mit dieser Dichte arbeiten; hierbei sind Programme, die mit ATARI BASIC, PILOT, LOGO, OSS BASIC XL, MAC/65, C/65 und anderen erstellt wurden, eingeschlossen. Natürlich werden auch Programme, die Sie selbst geschrieben haben und bei denen Sie sich keine Gedanken über die Größe und den Aufzeichnungstyp gemacht haben, einwandfrei unter DOS XL laufen.

Was funktioniert nicht unter doppelter Schreibdichte ? Programme, die erwarten, daß jeder Sektor 128 (oder 125) Bytes enthält, solche, die selbstbootend sind und solche, die spezielle Kopierschutztechniken eingebaut haben. Natürlich auch Programme, die das DOS vollkommen umgehen. Wie oben bereits erwähnt, befinden sich solche Programme sowieso auf selbst-bootenden Disketten und man muß nichts spezielles tun, um diese zum Laufen zu bringen.

Auf jeden Fall führen Sie bitte folgende Schritte durch, wenn Sie sich entschieden haben, welche Version Sie verwenden wollen:

1. Schließen Sie Ihre Diskettenstation und alle anderen Peripheriegeräte, die Sie verwenden wollen nach den Angaben des(der) Hersteller an.
2. Schalten Sie Ihre Peripheriegeräte und den TV oder Monitor ein.
3. Legen Sie die DOS XL Master Disk (entweder mit der Seite, die die einfache Schreibdichte oder mit der, die die doppelte Schreibdichte enthält) in die Diskettenstation 1.
4. Stecken Sie die BASIC-Cartridge bei den 400er und 800er Modellen in den Slot (oder Booten Sie ohne Drücken von OPTION bei den XL-Modellen), um mit dem BASIC arbeiten zu können.
5. Zuletzt schalten Sie bitte Ihren Computer ein.

Es findet ein Zugriff auf die Diskette statt und nach kurzer Zeit erscheint die Copyrightmeldung von DOS XL am oberen Rand des Bildschirms. Daraufhin erscheint nach und nach eine Begrüßung, die mit "Welcomes to DOS XL..."beginnt. In der letzten Zeile des STARTUP-Files steht lediglich "MENU". Diese Anweisung veranlaßt das DOS XL das DOS XL-Menü einzuladen. Nach ein paar Sekunden ist das Menüprogramm vollständig eingeladen, der Bildschirm wird gelöscht und das DOS XL-Menü erscheint.

Der erscheinende Begrüßungstext wird von einem speziellen File (STARTUP.EXC) erzeugt. In Kapitel 8.3 wird beschrieben, wie man diese Mitteilungen ändern oder unterdrücken kann.

2.2 Das Auswählen von Menü-Optionen

Nun sollte das DOS XL Menü auf dem Bildschirm stehen und wie folgt aussehen:

```
DOS XL MENU version 2.30
  copyright (c) 1983 OSS,Inc.

Files on Disk       Protect Files
To Cartridge        Unprotect Files
Copy Files          Rename Files
Duplicate Disk      Save Binary
Erase File          Load Binary
Initialiye Disk     Go to Adress
Xtended Command     Quit to DOS XL

Enter your selection.
```

Anmerkung: Sollte die Versionsnummer, die im Menü Ihrer Masterdisk angezeigt wird 2.31,2.32,etc...lauten, so ist das kein Grund zur Panik. Bei OSS dient die letzte Zahl der Revisionsnummer als ein Anhängsel, das mitteilt, daß keine größeren Veränderungen vorgenommen wurden, sondern nur kleinere Fehler früherer Versionen ausgebessert wurden.

Die Anfangsbuchstaben in den 12 Kommandos des Menüs sind in inverser Schrift dargestellt und zeigen damit an, daß man zum Anwählen dieses Punktes den korrespondierenden inversen Buchstaben drücken muß. Um z.B.die Option "RENAME"(Umbenennen) anzuwählen, wäre also der Buchstabe "R" zu betätigen.
Danach sind bei den meisten Menüpunkten noch weitere Angaben zu machen, die bei Bedarf durch Fragen angefordert werden.

Man muß nicht jedesmal, wenn das DOS XL nach einem Filenamen fragt, die Nummer der Diskettenstation und nicht einmal das D: angeben, sondern wenn nichts anderes angegeben wird, vermutet das System, daß Drive 1 gemeint ist.

Achtung: Generell darf man beim Arbeiten mit Cartridges, wie dem ATARI BASIC oder dem OSS BASIC XL die Diskspezifikation nicht weglassen. Außer, wenn man mit ~~dem DOS~~ XL oder dem CP arbeitet, muß man meist die Disknummer und den ganzen Filenamen und evtl.,so wie in ATARI BASIC, diesen in Anführungszeichen setzen.

2.3 Die Files auf der DOS XL Master Disk.

Um sich zu vergewissern, ob die DOS XL Master Disk vollständig ist, sollte man sich die Filenamen einmal auflisten lassen. Es sollte also jetzt das DOS XL Menü am Bildschirm stehen und mit folgenden Worten eine Eingabe angefordert werden:

Enter your selection.

Das erste Kommando des Menüs lautet: Files on Disk(Disketteninhaltsverzeichnis), dieses Kommando hat die Aufgabe alle auf der Disk befindlichen Programme aufzulisten. Vergewissern Sie sich bitte, daß sich die Master Disk noch in Laufwerk 1 befindet und betätigen Sie die Taste F.

Daraufhin fragt das Menü nach dem "Filespec"(File-Spezifizierung).
Hier einfach Return eingeben, und folgende Auflistung sollte am Bildschirm angezeigt werden(Die Zahlen können variieren):

```
* DOS     SYS 046
* DOSXL   SUP 046
* MENU    COM 025
* CLRDSK  COM 023
* COPY    COM 075
* DUPDBL  COM 011
* INIT    COM 006
* RS232   COM 001
* SDCOPY  COM 086
* VERIFY  COM 001
* SYSEQU  ASM 022
  STARTUP EXC 003
160 FREE SECTORS
```

Sollte die Auflistung im Prinzip so wie oben aussehen, die Nummern jedoch sind nur etwa halb so groß, dann haben Sie die Seite mit der Version der doppelten Schreibdichte gebootet. Konkret: Steht in der ersten Zeile:

```
* DOS     SYS 023
```

dann haben Sie mit Sicherheit momentan doppelte Schreibdichte. Die Nummer auf der rechten Seite entspricht nämlich der Größe des jeweiligen Files in Sektoren. Da die Sektoren unter doppelter Schreibdichte etwas mehr als das Doppelte als in einfacher Schreibdichte beinhalten, ist es logisch, daß die Files auch ~~nur etw~~a halb so lang sind. Beachten Sie auch wieviele freie Sektoren Sie hier mehr haben; das war ja wohl auch der Grund warum Sie sich ein Laufwerk mit doppelter Schreibdichte gekauft haben.

Jedes der Files hat einen primären Namen und einen Anhang(Extender), das File COPY.COM hat z.B.COPY als Namen und COM als Extender. Beachten Sie, daß in der Auflistung der Punkt nicht mit abgebildet wird. Dafür erscheinen aber einige Leerzeichen, um die Lesbarkeit zu erhöhen. Auf diese Weise dürfen Sie aber niemals ein File spezifizieren, sondern wie gehabt keine Leerstellen lassen und den Punkt zwischen eigentlichem Filenamen und dem Anhang verwenden.

Außer dem File STARTUP.EXC steht allen Files ein Sternchen voran, was anzeigt, daß ein File überschreib- und löschgeschützt wurde und zwar mit dem Kommando "Protect Files"(Schütze Files) des DOS Menüs. Diese Methode kann und sollte man anwenden, um auch eigene Files vor unbeabsichtigtem Löschen durch verschiedene andere DOS-Kommandos zu bewahren(mehr darüber in Kapitel 3.9).

2.4 Erstellen einer Sicherungs oder Arbeitskopie der DOS XL Master Disk

Jetzt, da Sie die DOS Master Disk erfolgreich eingeladen haben, sollten Sie gleich eine Sicherheitskopie anfertigen, so daß Sie jederzeit, falls einmal etwas mit Ihrer Master Disk schiefgehen sollte, die Programme wieder auf die Master Disk(oder eine andere) kopieren können.
Zuerst aber kleben Sie bitte einen Schreibschutzaufkleber über die Aussparung(en) an der Disk, sofern dort nicht schon einer angebracht ist, welcher die gesamte Diskette vor Be-und überschreibungen schützt. Diesen Schutz sollte man auch an seinen übrigen Programmdisketten anbringen, sofern man Sie nicht dauernd beschreiben muß.

Vom DOS Menü aus mit seinen 12 Kommandos gebe man auf die Aufforderung:

Enter your selection"

den Buchstaben "D" ein, welcher die Abkürzung zum Anwählen der Option "Duplicate Disk"(Vervielfältige Disk)ist. Nun wird man gefragt:

Double density ?

Jetzt hängt es davon ab, welche Zahlen beim Auflisten der Files rechts von den Programmnamen erschienen waren. Steht dort bei DOS.SYS 046, dann ist Single density mit der Eingabe von "N" >RETURN< anzuwählen.

Analog dazu ist bei doppelter Schreibdichte natürlich mit "Y" >RETURN< zu antworten.

Nun wird, abhängig von der getroffenen Wahl entweder das Programm DUPDSK.COM (Abkürzung für Vervielfältige Disk) oder DUPDBL.COM (Abk. f. Vervielfältige Disk in doppelter Schreibdichte) eingeladen und die folgende Anforderung wird am Bildschirm angezeigt:

Source disk (1,2,3,4) :
(Quelldiskette)

Normalerweise liegt diese im Laufwerk 1,
daher muß die Eingabe 1 >RETURN< lauten.

Auf die nächste Frage:

Destination disk (1,2,3,4) :
(Zieldiskette)

antwortet man mit der entsprechenden Zahl (1 bei nur einem Laufwerk).

Anmerkung: Wenn Sie 2 Laufwerke haben, dann lesen Sie bitte in Kapitel 7 nach, wie man diese für verschiedene Anwendungen konfiguriert. Sobald Sie dies erledigt haben, können Sie auch mit 2 Diskettenstationen arbeiten, was die Verarbeitungsgeschwindigkeit natürlich erhöht.

Die folgende Anforderung:

Format destination disk (Y/N):
(Zieldiskette formatieren ?)

ist analog zu oben zu beantworten. Die meisten neuen Disketten sind unformatiert, d.h. sie sind nicht darauf vorbereitet, irgendwelche Daten zu ordnen, so daß man, wenn man Daten auf eine frische Diskette kopieren möchte, diese zuerst formatieren lassen muß.

Nun bittet das Kopierprogramm:

Insert source disk into drive 1
And hit RETURN when ready
(Legen Sie die Quelldiskette in LW 1
und drücken Sie RETURN, wenn Sie soweit sind)

Da sich die Master Disk im Moment wohl noch in LW 1 befindet, drücken Sie lediglich >RETURN<.

Dannach wird die ~~BUSY~~-Lampe am Laufwerk aufleuchten, das Laufwerk anlaufen und das Kopierprogramm mitteilen:

Reading source disk
(Quelldisk wird gelesen)

Nach einer Weile wird man durch:
Insert destination disk into drive 1
And press RETURN when ready
(Legen Sie die Zieldskette in LW 1
und drücken Sie RETURN, wenn Sie soweit sind)
zu neuen Aktionen aufgefordert.

Zu diesem Zeitpunkt hat das Kopierprogramm soviel wie möglich in den Speicher geladen. Entnehmen Sie jetzt bitte die Master Disk, legen Sie eine frische Diskette ein und drücken Sie >RETURN<.

Hierauf wird das Programm antworten:
Formatting destination disk
(Zieldiskette wird formatiert)
und nach kurzer Zeit:
Writing destination disk
(Zieldiskette wird beschrieben)

Meistens aber sind die Informationen auf einer Diskette zu zahlreich, um auf einmal in den Speicher des ATARI zu passen, so daß man die Quell-und Zieldiskette entsprechend den Anweisungen mehrmals einlegen und entnehmen muß, bis die vollständige Duplizierung ausgeführt ist. Folgen Sie also den Bildschirmanweisungen genau, bis folgende Meldung erscheint:

Copy same disk again (Y/N)
(Gleiche Disk noch einmal kopieren?)
Tippen Sie jetzt "N" für Nein ein und >RETURN<.

Die nächste Meldung erscheint:
Hit RETURN for Menu
(Mit RETURN zum Menü)

Ganz klar: mit RETURN kommt man zurück ins Menü des DOS XL. Nun ist der ganze Kopiervorgang erfolgreich abgeschlossen. Kleben Sie wieder einen Überschreibschutz über die Aussparung an der Diskette, kennzeichnen Sie diese entsprechend und bewahren Sie sie an einem sicheren Ort auf.

2.5 Das Einlegen der Cartridge

Sollten Sie eine Cartridge einlegen wollen, so wie BASIC (nur bei 400er,800er Serie), BASIC XL oder ACTION!, gibt es auch hierfür ein DOS XL-Kommando.

Für 400er/800er Serie: Sollte sich die Cartridge nicht schon im Slot befinden, sollten Sie jetzt den ATARI ausschalten und diese einstecken. Nun den Computer wieder einschalten und, wenn die Diskettenstation eingeschaltet und die DOS XL Master Disk eingelegt ist, findet man sich nach kurzer Zeit im DOS XL Menü wieder.

Um in die Cartridge zu kommen ist das Kommando "To Cartridge" (Zum Modul) mit
Eingabe von "T" anzuwählen.

Selbstverständlich muß man bei den XL-Modellen keine Cartridge einlegen, sondern gelangt nach Eingabe von "T" in das eingebaute BASIC. Nun können Sie in BASIC (oder einer anderen Sprache) programmieren.

Wenn Sie die ATARI-BASIC-Cartridge verwenden(oder im BASIC der XL-Modelle programmieren), schauen Sie bitte im Kapitel 4 nach, in dem beschrieben wird, wie man aus dem BASIC wieder Zugriff auf die Diskette erlangt. Für jene, die schon mit dem ATARI DOS 2.0 zu tun hatten, sei gesagt, daß der Zugriff genau wie dort verläuft.

Kapitel 3: Das DOS-Menü

Das DOS XL-Menü wurde erstellt, um eine einfache Bedienung bei gleichzeitigem Zugriff auf das ganze System zu ermöglichen.

Für diejenigen von Ihnen, die vorher mit dem ATARI-DOS gearbeitet haben, erscheint im folgenden eine Zusammenstellung der Unterschiede zwischen DOS XL und dem ATARI-DOS:

1. Das Einladen des DOS - Das ATARI-DOS muß jedesmal, wenn man aus dem BASIC oder einer Cartridge ins DOS zurückkehren will, neu von Diskette eingeladen werden. Da dies ziemlich umständlich und langsam ist, wurde es so eingerichtet, daß das DOS XL ständig im Speicher bleibt, sofort zugriffsbereit ist und trotzdem nur ca.2000 Bytes zusätzlich von dem normalen Speicher benötigt (Dies kann durch Benutzen der OSS SuperCartridge wieder ausgeglichen werden). Sollte man auch diese 2 K benötigen, so muß man den Kommandoprozessor(CP) verwenden, der diesen zusätzlichen Speicherplatz nicht benötigt.

2. MEM SAV - Das ATARI-DOS unterstützt das MEM.SAV-File, in dem man vor einem Zugriff auf das DOS, das im Moment im Speicher befindliche Programm abspeichern kann, um später daran oder damit weiterzuarbeiten. DOS XL unterstützt dies nicht, da es kein MEM.SAV benötigt. Das Programm im Speicher bleibt auch nach einem DOS-Zugriff vollständig erhalten.

3. Kommando per Tastendruck - Bei DOS XL muß man lediglich einen Buchstaben drücken, um ein bestimmtes Kommando anzuwählen. Im Gegensatz dazu ist es beim ATARI-DOS notwendig, nach einer Eingabe noch die >RETURN<-Taste zu drücken.

3.1 Das Eingeben von Kommandos

Wann immer man in das DOS XL-Menü kommt, erscheint das auf Seite 11 abgebildete Menü,
das mit der Mitteilung
"Enter your selection" (Geben Sie Ihre Wahl ein) eine Auswahl anfordert.

An dieser Stelle muß man den Anfangsbuchstaben der gewünschten Option eingeben; sollten noch zusätzliche Informationen zu der angewählten Aktion notwendig sein, so fragt das DOS diese ab.

Außer wenn das DOS ein Hilfsprogramm einlädt, kann man eine Auswahl jederzeit mit der >ESC<-Taste abbrechen. Sollte das Programm gerade ein Hilfsprogramm einladen(wie z.B.bei Copy Files(Kopiere Files), Duplicate Disk(Dupliziere Diskette) oder Initialize Disk (Formatiere Diskette),dann ist dies nur mit >SYSTEM RESET< zu unterbrechen.

In den folgenden Unterkapiteln werden die einzelnen Befehle des DOS-Menüs in alphabetischer(nicht in der Reihenfolge des Menüs) behandelt.

3.2 C - Copy Files (Kopiere Files)

Entspricht im CP: COPY

Das COPY-Kommando ermöglicht es, einzelne Files auf andere Disketten oder zu anderen Programmen auf der gleichen Diskette zu kopieren. Das "Copy Files"-Kommando ist also hilfreich, wenn es darum geht, eines oder mehrere Files von einer auf eine andere Diskette zu kopieren. Sollte man jedoch vorhaben mehrere oder alle Files auf einer Diskette zu kopieren, so ist die Funktion "Duplicate Disk" (Dupliziere Diskette) schneller.

Aus dem Menü erreicht man den Kopiermodus durch Eingabe von "C". Nun prüft das DOS, ob sich das File "COPY.COM", welches den Filetransfer durchführt, auf der in Laufwerk 1 liegenden Diskette befindet. Wenn nicht, so erscheint die entsprechende Meldung:

Copy Files
Insert MASTER disk and hit RETURN
(Legen Sie die MASTER Disk ein und drücken Sie RETURN)

Sollte diese Mitteilung am Bildschirm erscheinen, so entnehmen Sie bitte Laufwerk 1 die aktuelle Disk, legen Sie die DOS XL-Masterdiskette ein und drücken >RETURN<.

Dannach erscheint folgendes:

Copy Files:
From file:
(Von File)

Jetzt mit dem/den entsprechenden Filenamen antworten; will man z.B. das File PROGR1 in Laufwerk 1 auf eine andere Disk kopieren, so muß die Eingabe D1:PROGR1 lauten.

Beachten Sie bitte, daß Sie hier auch Wildcards verwenden dürfen, um z.B. alle Files mit einem bestimmten Extender oder alle Files die mit einer bestimmten Buchstaben-oder Zahlenfolge beginnen oder enden, zu kopieren. Sollten Sie auch hier alle Files auf einer Disk auf eine andere kopieren wollen, so geben Sie einfach Dn: ein(wobei n für die Nummer der Diskettenstation(en) steht).

Hierauf fragt das COPY-Hilfsprogramm:

To file:
(Zum File)

Hier bitte analog zu oben mit dem Zielfilenamen antworten, wobei man meistens ein oder mehrere Programme von einer Disk auf eine andere kopieren will, ohne den Namen des Files zu ändern.

Sollte das der Fall sein, müssen Sie hier auch nur mit Dn: antworten, wobei "n" wiederum die Nummer für die Zielstation ist. Im oberen Beispiel muß man dann natürlich, wenn man nur eine Station besitzt auch D1: eingeben.

Die nächste Frage lautet:

Single Drive ?
(Nur eine Station?)

Wenn Sie also, wie gesagt nur eine Station besitzen oder im Moment nur mit einer arbeiten, beantworten Sie diese Frage bitte mit "Y" >RETURN< (Im anderen Fall natürlich mit "N").

Das COPY-Programm wird nun von der Diskette geladen und fordert auf:

Insert disk(s) to be copied
and hit RETURN when ready
(Legen Sie die Diskette(n) zum
Kopieren ein und drücken Sie
RETURN, wenn Sie soweit sind.)

Also: Entnehmen Sie die Master Disk (wenn sich das zu kopierende Programm auf der Masterdisk befindet dann natürlich nicht), legen Sie die Diskette mit dem Quellprogramm ein(wenn Sie mehrere Stationen benutzen, die Zieldiskette in die betreffende Station, zu der kopiert werden soll) und drücken Sie >RETURN<.

Vor dem Kopieren der einzelnen Files wird man gefragt:

Copy
Dn:filename
to Dn:filename,

was je nach Wunsch entweder mit "Y" für ja oder mit "N" für Nein beantwortet werden muß.

Sollten Sie sich entschlossen haben ein bestimmtes File nicht zu kopieren, so erscheint eine entsprechende Mitteilung am Bildschirm.

Nun wird das Quellfile in den Speicher eingelesen. Wenn Sie auf eine andere Diskette im gleichen Laufwerk kopieren, werden Sie gebeten jetzt die Zieldiskette einzulegen:

Insert 'to' disk and hit RETURN
(Legen ~~Sie die~~ "Nach"-Diskette
ein und drücken Sie RETURN)

Sollte das Zielfile bereits vorhanden sein, wird gefragt:

'To'file already exists
OK to overwrite?
(Das Zielfile existiert bereits
Soll es überschrieben werden?)

Dies natürlich auch wieder entweder mit "Y" oder "N" beantworten.

Sollte das Zielfile vorher schon mit dem Kommando "Protect files" (Schütze Files) mit einem Überschreibschutz versehen worden sein, d.h. ist es in der Directory-Auflistung mit einem vorangestellten Sternchen gekennzeichnet, so kann das COPY-Hilfsprogramm dieses nicht überschreiben. Der Schutz muß erst mit dem Kommando "Unprotect Files" (Entschütze Files) wieder rückgängig gemacht werden.

Es wird immer so viel wie möglich von dem Quellfile in den Speicher eingelesen; sollte dieses zu lang sein, um vollständig in den Speicher zu passen und man nur mit einer Station kopiert, so wird man mit:

Insert 'from' disk and hit RETURN

(Legen Sie die Quelldiskette ein und drücken Sie RETURN)
wiederun aufgefordert, die Quelldiskette einzulegen;

achten Sie auf die weiteren Anfragen des Kopierprogrammes bis der gesamte Kopiervorgang beendet ist.

Ist ein File erfolgreich kopiert worden, so erscheint eine entsprechende Meldung am Bildschirm. Nach erfolgreichem Kopieren des letzten Files erscheint:

Hit RETURN for menu.

Mit Eingabe von >RETURN< kommt man also zurück ins Menü.

Das "Copy Files"-Kommando sollte nicht zum kopieren von Disketten in einfacher auf Disketten mit doppelter Schreibdichte verwendet werden, wenn man nur eine Station besitzt. Sehen Sie bitte hierzu im Kapitel 7 nach. Im speziellen wird das Kopieren von einfacher auf doppelte Schreibdichte in Kapitel 7.3 behandelt. Weiterhin interessant zu diesem Thema: Kapitel 6.16 und das ganze Kapitel 7, das Sie am besten vorher ganz lesen sollten.

3.3 D - Duplicate Disk (Vervielfältige Diskette)

im Kommando Prozessor: DUPDSK
DUPDBL

Dieses Kommando ermöglicht es, schnell den gesamten Inhalt einer Diskette auf eine andere zu kopieren. Sollten Sie nur eines oder ein paar Files von einer auf eine andere Diskette kopieren wollen, oder sollten sich schon einige der Files der Quelldiskette bereits auf der Zieldiskette befinden, ist es praktischer das "Copy Files"-Kommando zu verwenden.

Der "Duplicate Disk"-Befehl schreibt vollkommen neue Informationen auf die Zieldiskette, so daß alle dort existierenden Files vernichtet werden. Wählen Sie also die Zieldiskette sorgfältig aus, damit Sie keine Programmverluste haben.

Der "Duplicate Disk"-Befehl wird mit der Eingabe von "D" angewählt, sodann erscheint folgende Frage:

Duplicate Disk
Double density?
(Vervielfältige Disk
Doppelte Schreibdichte?)

Wenn die Quelldiskette in einfacher Schreibdichte formatiert wurde, so ist hier mit "N" zu antworten, entsprechend bei doppelter Schreibdichte natürlich mit "Y".

Das Duplicate Disk-Hilfsprogramm wird in den Speicher geladen und die nächste Mitteilung lautet:

Source disk (1,2,3,4)
(Quelldiskette(in LW 1,2,3,4)

Normalerweise kopiert man von Laufwerk '1' aus,d.h.die Eingabe muß 1 lauten.

Destination Disk (1,2,3,4)
(Zieldiskette (in LW 1,2,3,4)

Wenn Sie nur ein Laufwerk besitzen oder nur eines verwenden wollen, geben Sie hier '1' ein, ansonsten die gewünschte Diskstation-Nummer.

Format ~~destination~~ disk(Y/N)
Zieldiskette formatieren)

Ist die Zieldiskette eine frische Diskette aus der Schachtel, dann 'Y' eingeben.

Weiter geht es mit:
        Insert source disk into drive 1
        and hit RETURN when ready
        (Quelldiskette in LW 1 legen)
      und RETURN drücken, wenn Sie soweit sind

oder mit:
        Insert source disk into drive 1
        Insert destination disk into drive n
        and hit RETURN when ready
        (Quelldiskette in LW 1 legen)
     Zieldiskette in das betreffende LW legen und RETURN drücken...

Verfahren Sie bitte wie beschrieben. Daraufhin wird die BUSY-Lampe aufleuchten und die Meldung:
        Reading source disk
    (Quelldiskette wird gelesen)

Wenn die Quell-und die Zieldiskettenstation identisch sind, so erscheint folgende Aufforderung:
        Insert destination disk in drive 1
        And hit RETURN when ready
        (Legen Sie die Zieldiskette in LW 1
        und drücken Sie RETURN...

Nachdem Sie dies ausgeführt haben, wird die Zieldiskette formatiert:
        Formatting destination disk
        (Zieldiskette wird formatiert)
und nach einer kurzen Weile erscheint:
        Writing destination disk
        (Zieldiskette wird beschrieben)

Wie im vorigen Kapitel auch wird auf einer Diskette meist mehr Information stehen als auf einmal in den Speicher des Computers paßt. Daher bitte den Mitteilungen am Bildschirm folgen und die Disketten dementsprechend wechseln, bis folgende Meldung erscheint:
        Copy same disk again (Y/N)?
    (Geiche Disk nocheinmal kopieren?)

Wenn Sie hier mit "Y" antworten beginnt der Prozess von Neuem, wenn nicht dann erscheint:
        Hit RETURN for menu
        (Mit RETURN zum Menü)
Um zum DOS XL Menü zurückzukehren, hier also >RETURN< eingeben.

3.4 E - Erase Files (Files löschen)

Entspricht im CP: ERAse

Das "Erase Files" Kommando erlaubt es eines oder mehrere Files auf einer Diskette zu löschen. Dieses Kommando sollte man mit Vorsicht anwenden, da es nicht so einfach ist, ein einmal gelöschtes File wieder zu reaktivieren, wenn es überhaupt möglich ist.

Sollten Sie das "Erase Files"-Kommando bei einem schreibgeschützten File (also bei einem, vor dem in der Directory-Auflistung ein Sternchen steht) anwenden wollen, so wird das sicherlich nicht funktionieren. Hierbei erscheint die Fehlermeldung "File Protected" (File schreibgeschützt); zuerst muß der Schreibschutz durch den "Unprotect Files"-Befehl (Entschütze File) rückgängig gemacht werden. Merken Sie sich aber, daß das Schützen von Files ein nützliches Vorsorgemittel vor unbeabsichtigtem Löschen ist.

Aus dem DOS XL-Menü gelangt man durch Eingabe von "E" in den Löschmodus, in dem die erste Meldung lautet:

Erase Files
Filespec to erase:
(Files löschen)
(Filename, des zu löschenden Files)

die man mit dem entsprechenden Filenamen des zu löschenden Files beantwortet. Wenn Sie mehrere Files auf einmal löschen wollen, dürfen hier auch wildcards verwendet werden, um z.B. alle Files mit dem Extender .BAS zu löschen, wäre hier *.BAS einzugeben.

Zur Sicherheit erfolgt hier noch einmal eine Anfrage:

Are you sure?
(Sind Sie sicher?)

Haben Sie den richtigen Filenamen eingegeben, so antworten Sie bitte mit "Y", andernfalls wird mit "N" der Löschvorgang abgebrochen. Haben Sie mit "Y" geantwortet, so werden jetzt alle Files, die mit der angegebenen Filespezifikation übereinstimmen von der Diskette gelöscht.

Mit RETURN gelangt man auf die folgende Meldung:

Hit RETURN for menu

zurück ins DOS XL-Menü.

3.5 F - Files on Disk (Disketteninhaltsverzeichnis)

im CP: DIRectory

Das "Files on Disk"-Kommando liefert eine Auflistung aller, sich auf der Diskette befindenden Programme.

Diese Option mit "F" anwählen und die gewünschte Diskette in das entsprechende Laufwerk einlegen:

Files on disk
(Disketteninhaltsverzeichnis)

Drückt man nur >RETURN< , werden alle auf der Diskette befindlichen Programme angezeigt.

Gibt man hier jedoch einen Filenamen (oder mehrere bei Verwendung der Wildcards) an, wird das DOS XL veranlaßt auf der Diskette nach den betreffenden Programmen zu suchen und diese am Bildschirm anzuzeigen.

Filename:	Aufgelistete Files:
GEORGE	Das File "GEORGE", wenn es sich auf der Disk befindet.
JUNK.SAV	Das File mit dem primären Namen "JUNK" und dem Extender "SAV", wenn es auf der Disk ist.
AB?	Jedes File, das keinen Extender hat, drei Zeichen lang ist und dessen erste beiden Buchstaben A und B sind (z.B.ABC, ABX, AB1 etc...)
CAT*	Jedes File das mit der Buchstabenfolge CAT beginnt und keinen Extender hat. (Hier würden CAT, CATCHER, CATTLE etc.erscheinen)
JOHN.??X	Jedes File, dessen primärer Name "JOHN" ist und einen aus drei Zeichen bestehenden Extender mit dem letzten Buchstaben "X" hat. (Bsp: JOHN.ABX, JOHN.XXX)

**.* Alle Files auf dieser Diskette. Diese Eingabe kann auch einfach mit >RETURN< abgekürzt werden.

D1: Alle Files auf der Disk in Laufwerk 1.

D2: Alle Files auf der Disk in LW 2.

3.6 G - Goto Address (Gehe zu Adresse...)

CP: RUN

Durch dieses Kommando wird die Kontrolle einem Maschinenprogramm übergeben, das sich bereits im Speicher des Computers ab einer bestimmten Addresse befindet und meist vorher mit dem Befehl "Load File" (File laden) eingeladen wurde.

"Go to Address" wird mit Eingeben von "G" angewählt, worauf die folgende Meldung erscheint:

Goto Adress
Address:
(Gehe zu Addresse)
(Addresse:)

Hier die gewünschte hexadezimale Addresse eingeben, zu der gesprungen werden soll; wenn ein Maschinenprogramm z.B: im Speicherbereich ab $5000 liegt (das vorangestellte Dollarzeichen kennzeichnet die nachfolgende Zahl als hexadezimal) das anzuspringende ML-Programm. In diesem Falle die Zahl 5000 (diese wird hier auch ohne vorangestelltes Dollarzeichen als Hexadezimal-Zahl behandelt) eingeben und >RETURN< drücken.

Vergewissern Sie sich, daß die hier eingegebene Zahl korrekt ist, da sonst in den meisten Fällen, in denen an der angewählten Speicherposition nicht das gewünschte ML-Programm steht, die Kontrolle über den Computer verloren ist und die Tastatur nicht mehr auf Eingaben reagiert. In manchen Fällen läßt sich die Kontrolle aber durch Drücken von >SYSTEM RESET< wiedererlangen. Meist jedoch ist man gezwungen, den Computer auszuschalten und neu zu booten.

Nun wird die Kontrolle an das Maschinenprogramm übergeben. Sollte sich am Ende des Programmes der 6502-Rücksprungbefehl RTS befinden, so erscheint:

Hit RETURN for menu

und man gelangt mit >RETURN< wieder ins DOS XL-Menü.

3.7 I - Initialize Disk (Diskette formatieren)

CP: INIT

Mit dem "Initialize Disk"-Kommando kann man Disketten formatieren, so daß man diese künftig zum Aufbewahren von Programmen und Daten vorbereiten kann. Sollten Sie eine bootbare Diskette, so wie die DOS XL-Masterdisk erstellen wollen, so wäre es praktischer das Komando "Duplicate Disk" zu verwenden, da dort die Zieldiskette vor dem Beschreiben auch formatiert werden kann.

Das "Initialize Disk"-Kommando schreibt völlig neue Informationen auf die ins Laufwerk eingelegte Disk, so daß evtl. darauf befindliche Programme unwiederbringlich gelöscht werden. Wählen Sie also die Disketten sorgsam aus, damit Sie keine Datenverluste haben.

Nach Auswählen von "I" prüft das DOS XL, ob sich das File "INIT.COM" auf der in LW 1 eingelegten Diskette befindet, sollte dies nicht der Fall sein werden Sie entsprechend aufgefordert die Master Disk einzulegen und dies mit >RETURN< zu bestätigen.

Hierauf wird das INIT-Programm geladen und die 4 Auswahlmöglichkeiten dieses Hilfsprogrammes angezeigt:

1. Format disk only (Diskette nur formatieren)
2. Format disk and write DOS.SYS (Diskette formatieren und mit DOS.SYS beschreiben).
3. Write DOS.SYS only (Nur DOS.SYS schreiben)
4. Exit to DOS XL. (Zurück zum DOS XL)

Achtung: normalerweise kann die Option "I" (INIT) nur verwendet werden, um Disketten in der selben Schreibdichte, in der die gebootete Master-Disk-Version initialisiert wurde, zu formatieren. Um eine andere Auswahl zu treffen, lesen Sie bitte Kapitel 7.

Eigentlich sollte man das "Duplicate Disk"-Kommando verwenden, um z.B. die DOS XL-Masterdisk zu kopieren, jedoch sind die Optionen 2 und 3 des "Initialize Disk"-Menüs ganz hilfreich, um aus einer bereits formatierten Diskette eine bootbare zu machen.

Verwenden Sie diese beiden Optionen aber nicht, wenn sie das File DOSXL.SYS gebootet und aktiviert haben, (was der Fall ist, wenn Sie den Instruktionen zur Verwendung des Extended memory DOS Systems, in Kapitel 8.1 gefolgt sind.

Sollten Sie eine in dieser Richtung modifizierte DOS-Konfiguration verwenden, so booten Sie bitte statt dessen die Original-Masterdisk (oder eine Arbeitskopie davon), bevor Sie die Optionen 2 und 3 aus dem "Initialize Disk"-Menü verwenden.

Anmerkung: Nach Verwenden der Auswahl 2, sollte das File DOSXL.SYS auf die betreffende Disk kopiert werden, wenn man die OSS SuperCartridge verwendet.

Haben Sie also den Wunsch eine bootbare Diskette zu erstellen, dann wählen Sie 2,
wenn es nur eine Datendiskette werden soll, dann 1 gefolgt von >RETURN<

Auf die Frage nach der/den Laufwerksnummer(n) bitte entsprechend, wie in den vorigen Kapiteln antworten.

Darauf erfolgt die Frage:

Option n drive n - Are you sure (Y/N) ?
Auswahl n Laufwerk n - Sind Sie sicher ?

Sind Sie mit Ihren bis hier gemachten Eingaben zufrieden, so bestätigen Sie hier mit "Y", andernfalls brechen Sie mit "N" ab.

Haben Sie mit "Y" geantwortet wird die Option im vorher angewählten Auswahlpunkt ausgeführt und wieder das INIT-Menü angezeigt.

Wollen Sie noch mehr Disketten formatieren, so wählen Sie bitte von Neuem, andernfalls gelangen Sie mit 4 zurück ins DOS XL-Menü.

3.8 L - Load Binary (Binär laden)

CP: LOAd

Das "Binary Load"-Kommando ermöglicht es, ein Binärfile von Diskette in den Computerspeicher einzuladen. Dieser Befehl kann also dazu verwendet werden binäre Objektfiles von Assembler-Programmen oder binäre Datenfiles, die von diesen Programmen genutzt werden, einzuladen. Das einzuladende File sollte vorher mit dem Kommando "Save Binary" (Binär speichern) oder einer anderen Äquivalenten Methode auf der Diskette abgelegt worden sein.

Verwenden Sie dieses Kommando nicht, um ATARI BASIC- oder BASIC XL-Programe in den Speicher zu laden, sondern verwenden Sie hierzu den LOAD-Befehl im BASIC. Bitte sehen Sie in Kapitel 4.6, für weitere Informationen hierzu, nach.

Durch Eingabe von "L" gelangt man in den Binär-Lade-Modus, wo man von der folgenden Meldung begrüßt wird:

Load Binary
Filename:
(Binär laden
Programmname),

worauf man mit dem Namen des einzuladenden Programmes antwortet.

Nach dem Einladen des entsprechenden Programmes erscheint die altbekannte Meldung und man gelangt mit >RETURN< zurück ins Menü.

3.9 P - Protect File (Schütze File)

CP: PROtect

In vielen Fällen wird man auf Disketten Programme haben, die man nicht mehr zu verändern braucht. Durch Verwendung des "Protect Files"-Kommandos und Auswahl von "P" kann man Programme, die man nicht unbeabsichtigt modifizieren, umbenennen oder gar völlig löschen will, schützen. Diese geschützten Files erscheinen in einer Disketteninhaltsauflistung mit einem vorangestellten Sternchen und müssen, falls man doch noch eine Veränderung vornehmen will, mit dem Befehl
"Unprotect Files" (Entschütze Files) wieder veränderbar gemacht werden.

Folgende Meldung erscheint am Bildschirm:

Protect Files
Filespec to protect:
(Files schützen)
(Name des/der zu schützenden Files)

Nun den Namen des zu schützenden Files eingeben. Durch Verwendung von Wildcards können mehrere oder alle Files auf einer Disk gleichzeitig geschützt werden. Mit RETURN gelangt man auch hier nach getaner Arbeit ins Menü.

3.10 Q - Quit to DOS XL (Verlassen zum DOS XL)

CP: kein äquivalent zu diesem Befehl.

Das "Quit to DOS XL"-Kommando wird verwendet, um vom DOS XL Menü in den DOS XL Kommando Prozessor zu gelangen. Obwohl man die gebräuchlichsten Kommandos des DOS XL auch bequem vom DOS XL-Menü aus erreichen kann, gibt es doch einige spezielle Optionen, die nur vom Kommando-Prozessor(CP) aus zugänglich sind.

Durch Eintippen von "Q" kommt man also in den CP, der sich mit der folgender Bildschirmausgabe meldet:

```
DOS XL - Atari version 2.30
  copyright (c) 1983 OSS, Inc.

D1:
```

Das "D1:" ist die Eingabeanforderung des Kommando-Prozessors, der nicht wie im Menü nur einen einzelnen Buchstaben, sondern eine ganze Kommandozeile als Auswahl erwartet; um z.B.das DUPDSK.COM-Hilfsprogramm einzuladen und zu aktivieren, müßte die Kommandozeile DUPDSK lauten.

Es gibt zwei Hauptkommandogruppen, die man auswählen kann, wenn das "D1:" erscheint: Interne und externe Kommandos. Für den User ist der einzige fühlbare Unterschied zwischen diesen beiden Gruppen, daß sich zur Ausführung eines externen Kommandos die DOS XL Masterdisk in Laufwerk 1 befinden muß, um das entsprechende Hilfsprogramm, das zu diesem Kommando gehört einladen und aktivieren zu können. In Kapitel 5 werden die internen und in Kapitel 6 die externen Kommandos näher besprochen.

Weiterhin ist es möglich eigene Kommandos zu gestalten oder die "Xtended Commands" (Extra-Kommandos)-Option zu verwenden. Zu diesem Thema sei auf Kapitel 10 verwiesen.

Verschiedene Funktionen sowie z.B. das Batch-prozessing (Programmzusammenhänge erzeugen) oder selbststartende Programme zu erstellen, sind nur vom CP aus erreichbar; dies wird in Kapitel 9 näher behandelt.

Um vom CP zurück ins DOS XL Menü zu gelangen, legen Sie bitte die Master Disk in Laufwerk 1 (auch wenn Sie mehrere Laufwerke angeschlossen haben, die Masterdisk muß immer in LW 1 liegen) ein und tippen

"MENU" >RETURN<.

Das DOS XL Menü wird daraufhin geladen und aktiviert.

3.11 R - Rename File (File umbenennen)

im CP: REName

Das "Rename File"-Kommando kann verwendet werden, um den Namen, der zu einem bestimmten File zugehörig ist, nach eigenen Vorstellungen zu verändern. Diese Kommando hat keinerlei Einflüße auf den Inhalt des eigentlichen Files, sondern ändert lediglich den Namen des Programmes in der Directory (dem Ihaltsverzeichnis) der aktuellen Diskette und erscheint in einer Auflistung, die man mit dem Befehl "Files on disk" erreichen kann unter dem neuen Namen.

Auch hier wird man mit dem Umbenennen keine Erfolge haben, wenn das File schreibgeschützt ist. Die Meldung "FILE PROTECTED" erscheint. In diesem Falle mit "Unprotect File" dieses entschützen und noch einmal versuchen.

Mit "R" wird die Umbenenn-Funktion aufgerufen und die Meldung:

    Rename File
    Old Name:
    (File umbenennen
    Alter Name ?)

erscheint, worauf man mit dem aktuellen Namen des umzubenennenden Files antwortet. Danach wird mit:

    New name:
    (Neuer Name ?)

nach eben diesem gefragt. Nun folgt noch eine Sicherheitsabfrage:

    Are you sure?
    (Sind Sie sicher?),

nach der, wenn man mit "Y" geantwortet hat, die Anforderung ausgeführt wird. Mit "N" kann hier abgebrochen werden.

Auch hier geht ~~es nach~~ vollendeter Arbeit wieder mit >RETURN< zurück zum Menü.

3.12 S - Save Binary (Binär abspeichern)

CP: SAVe

Die "Save Binary"-Anweisung ermöglicht es Teile des Computerspeichers auf Diskette abzulegen. Hiermit lassen sich also binäre Objektfiles von Assembler Programmen oder binäre Datenfiles, die von diesen Programmen benutzt werden auf Diskette abspeichern.

Mit diesem Kommando kann man keine ATARI-BASIC oder BASIC XL-Programme auf Disk schreiben. Hierzu ist der Befehl "SAVE" im BASIC zu verwenden. Mehr hierzu in Kapitel 4.12.

Sollte sich bereits ein schreibgeschütztes Programm mit dem von Ihnen hier ausgewählten Filenamen auf der aktuellen Diskette befinden, so erscheint die Fehlermeldung "FILE PROTECTED". Wenn Sie das File trotzdem überschreiben wollen, weil sie zum Beispiel die verbesserte Version sichern wollen, zuerst mit "Unprotect File" das Programm entschützen und dann noch einmal versuchen.

Nach Eingabe von "S" erscheint:

```
        Save Binary
       Filename:
        (Binär abspeichern
        Programmname),
```

wodurch man aufgefordert wird den entsprechenden Namen einzugeben. Will man also den Speicherbereich von $4000 bis $4100 (die vorangestellten Dollarzeichen zeigen an, daß es sich um hexadezimale Zahlen handelt) in dem File "FILE1.OBJ" auf der Diskette in Laufwerk 1 ablegen, so wäre "D:FILE1.OBJ" einzugeben.

Es hat sich als brauchbar erwiesen auf die Programmart durch den Extender hinzuweisen, wie z.B. OBJ oder COM. Im vorhergehenden Beispiel, zeigt der Anhang "OBJ", ~~daß es~~ sich um ein Maschinensprache-OBJektfile handelt. Der andere Extender, "COM", zeigt an, daß das Programm ein ML-File ist, das mit dem "Load Binary"-Kommando eingeladen und gestartet werden kann. (Auf der Masterdisk befinden sich ebensolche Files.

Nun wird nach der Startaddrese gefragt:

Starting address:,

worauf man mit der hexadezimalen Zahl antworten soll, die der ersten Speicherzelle des abzuspeichernden Programmes angehört. Im oberen Beispiel ist dies die Speicherstelle $4000, so daß die Eingabe 4000 lauten muß.

Auf:

Ending address:

antworte man bitte mit der letzten Speicherstelle, die noch zum ML-Programm gehören soll, hier also mit 4100.

An dieser Stelle wird der gewünschte Speicherbereich unter dem ausgsuchten Filenamen auf die Diskette geschrieben.
Mit >RETURN< gelangt man wieder ins Menü.

3.13 T - To Cartridge (Zum Modul/Zum BASIC)

CP: CARtridge

Mit diesem Kommando gelangt man in ein eingestecktes Modul oder in das eingebaute BASIC bei den XL-Modellen.

Wenn Sie die ATARI BASIC Cartridge verwenden oder mit dem in die XL-Modelle eingebauten BASIC arbeiten, beachten Sie bitte Kapitel 4 für Informationen über Kommandos, die es von dort aus ermöglichen auf Disketten zuzugreifen. Für diejenigen, die bereits mit dem ATARI DOS 2.0 gearbeitet haben, sei angedeutet, daß der Diskettenzugriff hier genauso verläuft.

Um dieses Kommando anzuwählen, bitte auf die Frage nach der Auswahl "T" eingeben, worauf sofort das vertraute READY des BASIC erscheint oder die Begrüßung des eingelegten Moduls. Sollte sich keine Cartridge im Slot befinden oder ohne BASIC gebootet worden sein, so erscheint die Fehlermeldung "NO CARTRIDGE" (Kein Modul)

Wenn Sie den "To Cartridge"-Befehl nach einem der folgenden Kommandos:
CopY Files (Files kopieren)
Duplicate Disk (Disk vervielfältigen)
Initialize disk (Diskette formatieren)
Xtended Commands (Extra Kommandos)
Load Binary (Binär laden)
anwählen, dann führt die Cartridge (oder das BASIC) einen Kaltstart aus, der jedes, sich im Speicher befindliche Programm löscht. Sollten Sie also aus dem BASIC ins DOS XL-Menü gehen wollen, so speichern Sie bitte vorher den Speicherinhalt auf Disk ab, was in ATARI BASIC oder BASIC XL durch den Befehl "Save" erreicht wird.

3.14 U - Unprotect Files (Files entschützen)

CP: UNProtect

Das "Unprotect Files"-Kommando ermöglicht es Files umzubenennen, zu löschen oder zu modifizieren, indem es einen evtl. Überschreibschutz entfernt. Schreibgeschützte Files erkennt man daran, daß in einer Disketteninhalts-Auflistung, z.B. mit "Files on Disk", vor dem entsprechenden Filenamen ein Sternchen abgebildet ist.

Wählen Sie also diese Option mit "U" aus und es erscheint:

```
        Unprotect Files
       Files to unprotect:
        (Entschütze Files
 Namen, der zu entschützenden Files)
```

Nun den Namen des gewünschten Files eingeben.

Wollen Sie mehrere Files auf einmal entschützen, so können auch hier Wildcards verwendet werden.

Die Informationen auf der Diskette werden jetzt entsprechend geändert und

mit >RETURN< gelangt man wie gehabt ins Menü zurück.

3.15 X - Xtended Command (Extra Kommandos)

CP: Kein Befehl hierfür

Mit Hilfe dieser Option wird ein Kommando an den Kommando Prozessor (CP) weitergegeben. Obwohl die meisten Kommandos auch aus dem DOS XL-Menü erreichbar sind, gibt es doch einige, die nur durch den CP aufgerufen werden können. Die Option "Xtended command" dient dazu durch den CP auf diese bestimmten Kommandos zugreifen zu können. Mehr über die Kommandos und Vorzüge des Kommando Prozessors in den Kapiteln 5 und 6.

Mit "X" also die Option anwählen und auf die Frage:

```
       Xtended Command
      Command:
       (Extra Kommando)
       (Kommando?)
```

mit dem entsprechend gewünschten DOS XL- Kommando antworten, das ausgeführt werden soll; wenn Sie z.B. die RS232-Schnittstelle initialisieren wollen, wäre jetzt RS232 einzutippen.

Viele der DOS XL Extra Kommandos, die durch die "Xtended Command"-Funktion ausgewählt werden können, müßen von der DOS XL Masterdisk geladen werden. Beachten Sie bitte, daß sich diese Diskette in LW1 befindet. (Beschreibungen über die externen Befehle in Kapitel 6).

Nun wird das gewählte Kommando geladen und ausgeführt;

danach mit RETURN zurück ins DOS XL-Menü.

Kapitel 4: ATARI BASIC und das DOS XL.

Wenn Sie die DOS Masterdisk booten, gelangen Sie nach einer Begrüßung in das DOS XL-Menü. In Kapitel 2.5 und in Kapitel 3.13 konnten Sie bereits etwas über das "T - To Cartridge" (Zum Modul)-Kommando lesen. Haben Sie also die DOS XL Masterdisk mit BASIC XL, mit dem eingestecktem BASIC-Modul oder mit dem, in die XL-Modelle eingebauten BASIC gebootet, so sollten Sie nach Anwählen von "T" die bekannte READY-Meldung des BASIC vor sich am Bildschirm haben.

Sollte dies nicht der Fall sein, haben Sie entweder vergessen die BASIC-Cartridge einzulegen, oder bei den XL-Modellen während des Einschaltens des Computers die OPTION-Taste gedrückt gehalten. Wenn dies so ist, schalten Sie einfach Ihren Computer aus, stecken das BASIC-Modul in den Slot (oder unterlassen es, die OPTION-Taste zu drücken (XL)) und schalten wieder ein.

Von nun an und in den folgenden Kapiteln wird vorausgesetzt, daß man sich im ATARI BASIC oder im BASIC XL befindet. Hier kann man jederzeit, wenn die READY-Meldung am Bildschirm erscheint, durch Eingabe von "DOS" in das DOS XL gelangen. Generell sollte sich das Programm, das sich im Speicher befindet auch nach einem Ansprung des DOS und Rückkehr aus diesem mit dem Kommando "T", nicht verändert haben. Lesen Sie aber bitte die Hinweise in Kapitel 3.13 durch, um ganz sicher zu gehen.

Aus dem Kommando Prozessor-Modus gelangt man durch Eingabe von "CAR" ins BASIC. Analog dazu kommt man auch aus dem BASIC mit dem "DOS"-Befehl zurück in den CP.

===

Beachten Sie bitte, daß wenn Sie durch den "Q"-Befehl aus dem Menü in den CP gegangen sind und daraufhin das "CAR"-Kommando verwendet haben, um weiter ins BASIC zu gelangen, hier nun ca. 2000 Bytes mehr zur Verfügung stehen, da nun auch der Speicherbereich, den vorher das MENü benötigte, für eigene BASIC-Programme nutzbar wird. Weiterführende Informationen auch über den BOOT-Prozess des DOS Xl sind in Kap.8 zu finden.

In den nun folgenden Unterkapitel werden die gebräuchlichsten BASIC-Befehle für Diskettenzugriffe in alphabetischer Reihenfolge behandelt. Diese Kommandos sollten im direkten Modus, also nach erscheinen von READY eingegeben werden.

4.1 CLOSE (Schließe)

Kommando : CLOSE

Zweck : Dieser Befehl schließt einen mit OPEN zuvor geöffneten Kanal oder ein geöffnetes File.

Anwendung : Im BASIC XL und ATARI BASIC.

Angaben : Kanal/File Nummer (1-7)

Beispiele : CLOSE #1
CLOSE #1OUTFILE

Beschreibung:

Nachdem ein Kanal geschlossen wurde kann man nicht mehr auf das für diesen Kanal geöffnete File zugreifen(wie z.B.mit PRINT,Input etc...).

Anmerkung: Ein mit OPEN für jede Art von OUTPUT geöffnetes File sollte immer geschlossen werden, bevor man die Diskette, auf der sich das geöffnete File befindet. aus der Diskettenstation herausnimmt. Durch Nichtbeachten dieser Regel passieren die häufigsten Disketten-Fehler.

Das ATARI BASIC interessiert es überhaupt nicht, wenn ein File, das gar nicht geöffnet war, geschlossen wird, so daß es praktisch ist, ein Unterprogramm oder ein ganzes Programm mit folgender Zeile abzuschließen:

```
999 FOR I+1 TO 7 : CLOSE #I : NEXT I
```

Allerdings schließen auch die beiden Befehle END und RUN alle Files, außer Kanal #0 für die Tastatur und den Bildschirm natürlich und kann ebenso verwendet werden.

4.2 ENTER (Einlesen)

Kommando : ENTER

Zweck : Dieser Befehl wird verwendet, um ein BASIC-Programm, das zuvor mit LIST auf die Diskette geschrieben wurde, einzuladen.

Anwendung : Im BASIC XL und ATARI BASIC.

Schreibweise: ENTER Filename

Angaben : Filename, des zu ENTERnden Programmes.

Beispiele : ENTER "D:PROGR1.LIS"
ENTER OVERLAYFILES$

Beschreibung:

Funktioniert, wie oben beschrieben. Beim einlesen in den Benutzer-Speicher, wird die Syntax jeder einzelnen Programmzeile, als ob diese von Hand eingegeben würde, überprüft und in die interne (tokenisierte) Form des BASIC umgewandelt.

Wenn ein Fehler in der Schreibweise entdeckt wird, so erscheint die betreffende Zeile mit der Fehlerstelle in inverser Schrift am Bildschirm.

Diese Zeile befindet sich aber auf jeden Fall im Speicher, so daß diese vor einem RUN korrigiert werden muß.

<u>Achtung:</u> ENTER löscht im Gegensatz zu <u>LOAD</u> <u>nicht</u> den Programmspeicher, sondern vermischt evtl. schon dort bestehende Programmzeilen mit den eingeladenen, wobei Zeilen mit gleicher Zeilennummer überschrieben werden. Daher sollte man vorher NEW anwenden. Auf der anderen Seite ist dies eine praktische Möglichkeit Programme aneinanderzuhängen oder verschiedene Unterprogramme zu einem Gesamtprogramm zu vereinigen.

4.3 GET (Hole)

Kommando : GET

Zweck : Das Einlesen eines bestimmten Bytes aus einem Datenfile.

Verwendung : BASIC XL/ATARI BASIC

Anwendungsweise: GET #fn,numerische variable
Kanalnummer (1-7) und Variable

Beispiele : GET #1,BYTE
GET #INFILE,WERT

Beschreibung:

Mit dem GET-Kommando kann man einzelne Bytes aus einem vorher mit OPEN geöffnetem File auslesen indem man die gleiche Kanalnummer verwendet.

Anmerkung: Die von GET in den Speicher geholten Daten, sollten vorher mit Hilfe des PUT-Befehles dort abgelegt worden sein.

4.4 INPUT (Eingabe)

Kommando : INPUT

Zweck : Dieser Befehl wird angewendet, um Daten von der angegebenen Kanalnummer oder der Tastatur zu erhalten.

Verwendung in : BASIC XL/ATARI BASIC

Anwendungsweise: INPUT #fn,VARIABLE(numerischer oder in STRING-Form)

Beispiele : INPUT #3,NAME$
INPUT #INFILE,WERT1,WERT2

Beschreibung:

Wenn man die INPUT-Anweisung ohne Kanalnummer verwendet, werden die Daten von der Tastatur erwartet. Hierbei erscheint ein Fragezeichen am Bildschirm, das die Eingabe anfordert. Bitte sehen Sie in Ihrem ATARI-BASIC oder BASIC XL-Handbuch für detaillierte Informationen nach.

Wird eine Kanalnummer verwendet, so werden die Daten von einem, zuvor erfolgreich mit OPEN geöffneten File, in ATASCII-Zeilen-Form erwartet. Ansonsten ist der Input-Vorgang in etwa mit dem bei der Tastatur identisch, d.h. eine String-Eingabe wird mit dem ATASCII-Zeichen für RETURN und eine numerische mit einem RETURN oder einem Komma beendet.

Anmerkung: Die INPUT-Anweisung kann generell nicht mehr als 127 Zeichen lesen. Wenn man eine Zeile auf die Diskette PRINTed, die man später wieder mit dem INPUT-Befehl einlesen möchte, so sollte man diese auf 127 Zeichen oder weniger beschränken.

4.5 LIST (Auflisten)

Kommando : LIST

Zweck : Das Auflisten eines sich im Speicher befindlichen Programmes am Bildschirm oder Ablegen in einem angegebenenen File auf der Diskette.

Verwendung : BASIC XL/ATARI BASIC

Anwendungsweise: LIST oder LIST Zeilennumer1,Zeilennumer2,
LIST Filename oder
LIST Filename,Zeilennummer1,Zeilennumer2

Beispiele : LIST "D2:PROGR.LIS"
LIST FILE$, 1000, 2000

Beschreibung:

Das LIST-Kommando ist wohl eines der meistgebrauchtesten BASIC-Anweisungen, von dessen Funktion des Auflistens von Programmzeilen (auch Teilen eines Listings durch Eingabe der Anfangs-und Endzeilennumer) und zum Speichern eines Programmes im LIST-Format auf Diskette, wohl die meisten BASIC-Anwender schon einmal Gebrauch gemacht haben.

Gibt man nach dem LIST-Befehl nur eine Zeilennummer an, so erscheint auch nur die eine Zeile am Bildschirm, oder es wird auch nur die eine Zeile auf Diskette geschrieben; ansonsten immer Anfangs-und Endzeilennummer durch Kommas getrennt, eingeben.

Anmerkung: Durch das LISTen einer bestimmten Anzahl von Zeilen kann man z.B.ein universelles Unterprogramm aus einem Gesamtprogramm herausnehmen, um es in eine anderes zu integrieren (Siehe auch Kapitel 4.2 ENTER).

4.6 LOAD (Laden)

Kommando : LOAD

Zweck : Das Einladen eines, vorher mit SAVE auf die Diskette geschriebenen Programmes in den Arbeitsspeicher

Verwendung : BASIC XL/ATARI BASIC

Anwendungsweise: LOAD Filename

Beispiele : LOAD"D:GAME.SAV"

Beschreibung:

Da sich die mit SAVE abgespeicherten Programme bereits im tokenisierten Code befinden, wird hier beim Einladen keine Syntax-überprüfung vorgenommen und evtl. Fehler werden erst bei der Programmausführung bemerkt.

Nachdem man ein Programm geladen hat, kann man dieses nach einer evtl. Modifikation entweder unter dem gleichen Filenamen, wodurch die vorige Version überschrieben wird, oder unter einem anderen Namen abspeichern, da sich das BASIC den zuvor zum Einladen verwendeten Filenamen nicht merkt.

Um ein Programm lediglich einzuladen und gleich darauf zu starten, wäre die Verwendung des RUN "D:Filename.EXT"-Kommandos arbeitssparender.

4.7 NOTE (Merke)

Kommando : NOTE

Zweck : Dieses Kommando merkt sich die aktuelle physikalische Position eines, innerhalb eines mit OPEN geöffneten Files für eine spätere Verwendung mit dem POINT-Kommando

Verwendung : BASIC XL/ATARI BASIC

Anwendungsweise: NOTE #fn,Variable1,Variable2 (In Variable 1 wird die aktuelle Sektornummer und in 2 der Byte-Offset des aktuellen Sektors eingetragen).

Beispiele : NOTE #1, SECTOR, BYTE
NOTE #INFILE, S, B

Beschreibung:

Zur Anwendung dieses Kommandos benötigt man generell tiefergehende Kenntnisse in BASIC-Programmierung und im Umgang mit Datenfiles.

Die 2ten Versionen von DOS XL sowie dem ATARI-DOS arbeiten nur mit sequentiellen Files, mit vorwärts gerichteter Verknüpfung und Filenummerüberprüfung, was die letzten drei Bytes eines physikalischen Sektors belegt. Es wird kein wahlloser Zugriff auf innerhalb eines Files liegende Sektoren ohne die Kommandos NOTE und POINT ermöglicht, so daß der Programmierer genötigt ist, ein File von Anfang an einzulesen und zu modifizieren, wenn er nicht mit den oben genannten Befehlen arbeitet.

Dank NOTE und POINT und der vorwärtsgerichteten Verknüpfung kann man sich aber den wahllosen Zugriff auf einzelne Sektoren innerhalb eines Files, oder einzelne eines noch zu schreibenden Files, selbst stricken.

NOTE merkt sich wie gesagt den aktuellen Disksektor und den Byte-Offset dieses Sektors eines momentan geöffneten Diskfiles, indem es die von dort gelesenen Informationen in vom User definierte Variablen ablegt. Natürlich muß der User dafür Sorge tragen, daß diese Informationen sicher aufbewahrt werden, bis sie für den POINT-Befehl wieder benötigt werden.

Das folgende Beispielprogramm ist nicht das Erschöpfendste, gibt aber dennoch eine Grundlage zum Verständnis von freiem Sektorenzugriff unter DOS XL und ATARI-DOS.

```
100 REM TEIL 1: FILE AUF DER DISK ERSTELLEN
110 DIM LINE$(150):REM EINE ANNEHMBARE GROESSE
120 DIM SECTOR(100),BYTE(100): REM ditto
130 OPEN #1,8,0,"D:TESTFILE":REM DAS NEUE FILE
140 FOR I=1 TO 100
150 PRINT "Zeilennummer eingeben";I;:INPUT LINE$
160 IF LEN LINE$=0 THEN 200
170 NOTE #1,SECTOR,BYTE
180 SECTOR(I)=SECTOR:BYTE(I)=BYTE
190 PRINT #1,LINE$:NEXT I
    200 REM NUN IST DAS FILE ERSTELLT.
210 CLOSE #1
220 MAXREC=I-1:REM WIRKLICH...PROBIEREN SIE ES AUS
    250 REM TEIL2: DER ZUGRIFF AUF DAS FILE
260 OPEN #1,12,0,"D:TESTFILE":REM FILE, DAS ERSTELLT WURDE

300 REM HAUPTPROGRAMM
310 PRINT "BITTE GEBEN SIE EINE NUMMER ZWISCHEN 1 UND";MAXREC
320 PRINT " EIN(OR 0 ZUM ABBRECHEN)
330 INPUT RECORD
340 IF RECORD=0 THEN END
350 SECTOR=SECTOR(RECORD):BYTE=BYTE(RECORD)
360 POINT #1,SECTOR,BYTE
370 INPUT #1,LINE$
380 PRINT "ZEILE ";RECORD;"==";LINE$
390 GOTO 300
```

Dieses Programm kann von hier aus direkt übernommen werden.
Wenn Sie das Programm laufenlassen, beachten Sie bitte, daß sich die eingegebenen Zahlen im Rahmen zwischen 1 und MAXREC befinden, da das Programm dies nicht überprüft.

Sollten Sie Schwierigkeiten mit dem wahlfreien Zugriff auf Sektoren haben, lassen Sie dieses Thema einfach weg, denn man muß dies als BASIC-Programmierer nicht unbedingt beherrschen. Sollten Sie schon an Computern gearbeitet haben, die einen wahlweisen Zugriff auf Disksektoren per Bestimmung der relativen Byteposition anwenden, sollten Sie erwägen das OS/A+ Version 4, das für ATARI-Computer und Diskettenstationen mit echter doppelter Schreibdichte (und mehr) geschrieben wurde, anzuwenden.

Um bei der Version 4 eine File-Positionierung vorzunehmen, muß man lediglich eine 24-bit Byte Nummer, die dem Offset für den Anfang eines Files entspricht, angeben. Dieses DOS ist nicht für Anfänger und Programmierer, die auf Kompatibilität mit dem ATARI DOS 2.x wertlegen, geeignet.

4.8 OPEN (öffne)

Kommando : OPEN

Zweck : Dieses Kommando bereitet ein File für den Zugriff vor und erteilt diesem eine Kanalnummer.

Anwendung in : ATARI BASIC und BASIC XL

Anwendungsweise: OPEN #fn,var1,var2,Filename
var1 - In/Output-Modus
4 - für Eingabe
6 - für Disketteninhaltsverzeichnis-Zugriff
8 - für Ausgabe
9 - für Anhängen
12 - für gleichzeitige Ein-und Ausgabe
var2 - Wert, abhängig vom Peripheriegerät

Beispiele :

```
OPEN #1,8,0,"D:NEWFILE"
INMODE=4 : INFILE=3
INPUT INFILE$
OPEN #INFILE,INMODE,0,INFILE$
```

Beschreibung :

Die OPEN-Anweisung erlaubt es, ein Diskfile (oder jedes andere Gerät für diesen Fall) mit einem Kanal zu verknüpfen, der dann speziell für dieses geöffnete File ansprechbar ist (z.B. bei PUT, GET, INPUT,PRINT,CLOSE).

Anmerkungen zur Anwendungsweise:

Für fn darf eine Zahl zwischen 1 und 4 stehen, da die 0 für den Bildschirm reserviert ist und in ATARI BASIC nicht verwendet werden kann (in BASIC XL ist die 0 allerdings erlaubt). Nachdem ein File mit einem Wert für fn geöffnet wurde, muß dieses File auch unter dem entsprechenden fn angesprochen werden.

Das var1 erlaubt den OPEN-Befehl in verschiedenen Modi zu verwenden, was aus der folgenden Tabelle zu ersehen ist:

Modus 4 : öffnet das angegebene File nur für Eingabe, d.h. man kann nur Daten von dem File einlesen.

Modus 6 : erlaubt den Zugriff auf das Disketteninhaltsverzeichnis der betreffenden aktuellen Diskette.

Modus 8 : ist das Gegenteil von Modus 4 und erlaubt nur die Ausgabe, d.h. Daten können nur in das betreffende File geschrieben werden. Siehe weiter unten für weitere Erklärungen.

Modus 9 : wird verwendet, um Daten an ein bereits bestehendes File anzuhängen, d.h. die neuen Daten werden ab der Stelle geschrieben, an der das ursprüngliche File endet.

Modus 12: öffnet das File für Ein-und Ausgabe gleichzeitig; hier können sowohl Daten geschrieben als auch empfangen werden.

Achtung: Nachdem ein File unter einem bestimmten Kanal für eine bestimmte Operation geöffnet wurde, kann man keinen zweiten Kanal für eine andere Operation für dieses File öffnen. Zwei geöffnete Files können nicht die gleiche Kanalnummer haben, wohl aber kann ein File für zwei verschiedene Kanäle geöffnet werden, allerdings wird das ein ziemliches Durcheinander ergeben. Also lassen Sie das besser.

Beachte: Wenn ein File für Ausgabe geöffnet ist (var1=8) und das angegebene File existiert noch gar nicht, so wird eines geschaffen. Sollte das File unter gleichem Namen schon einmal vorhanden sein, so wird dieses zerstört und von dem Neuen überschrieben.

Vorsicht: An ein File, das mit var1=12 für Ein-und Ausgabe geöffnet wurde, kann man unter DOS XL oder ATARI-DOS nichts anhängen.

Der Modus 6 kann z.B. vom BASIC aus benutzt werden, um die Files auf einer Diskette nachzusehen und eine Menüauswahl zu ermöglichen. Das folgende Programm stellt eine solche Möglichkeit bereit, wobei alle mit SAVE abgespeicherten Programme von LW 1 eingeladen werden können, die Programmnamen aber keine Extender haben dürfen.

```
100 OPEN #1,6,0,"D:*":DIM LN$(40)
110 FOR I=1 TO 20:INPUT #1,LN$
120 IF LN$(2,2)=" " THEN PRINT I,LN$(3,10):NEXT I
130 CLOSE #1:OPEN #1,6,0,"D:*"
140 PRINT :PRINT "WELCHES PROGRAMM SOLL GESTARTET WERDEN";
150 INPUT J:IF J>=I THEN GOTO 140
160 FOR I+1 TO J:INPUT #1,LN$:NEXT I
170 CLOSE #1:LN$(1,2)="D:"
180 RUN LN$(1,10)
```

Tippen Sie dieses kleine Hilfsprogramm ein, speichern Sie es mit D"MENU" ab, dann können Sie es jederzeit unter RUN"D:MENU" erreichen.

4.9 POINT (Zeige)

Kommando : POINT

Zweck : Dieses Kommando merkt sich die aktuelle physikalische Position auf einer Diskette für einen späteren Zugriff.

Anwendung in : ATARI BASIC und BASIC XL

Anwendungsweise: POINT #fn,var1,var2
- fn - Kanalnummer von 1 bis 7
- var1 - diese Variable gibt die Sektornummer an
- var2 - gibt den Byte Offset des Sektors an

Beispiele : POINT #1, SECTOR, BYTE
POINT #1,INFILE, S, B

Beschreibung :

Für diesen Befehl braucht man im allgemeinen ein tiefergehendes Wissen über BASIC und den Umgang mit Datenfiles, bevor man ihn voll ausnutzen kann.
(Siehe Kapitel 4.7 für weitere Informationen über den Byte Offset und die Speicherung innerhalb eines Sektors).

Das Kommando POINT gibt lediglich den aktuellen Disk Sektor und den zugehörigen Byte Offset des geöffneten Files an.

Ist der ausgesuchte Sektor momentan Bestandteil eines geöffneten Files unter einer bestimmten Kanalnummer und ist der Byte-Offset gültig, dann wird der nächste Zugriff, sei es Ein-oder Ausgabe, auf dieses File durch den dafür geöffneten Kanal an dieser Stelle durchgeführt. Der POINT-Befehl wird meist in Zusammenhang mit einer Aktualisierung eines Files unter Modus 12 eingesetzt.
In Kapitel 4.7 findet sich auch ein Beispielprogramm hierzu.

4.10 PRINT (Schreibe)

Kommando : PRINT

Zweck : Dieses Kommando gibt den ASCII-Wert des gegebenen Ausdrucks an das File oder den Bildschirm weiter.

Verwendung : ATARI-BASIC und BASIC XL

Anwendungsweise : PRINT (#fn;)exp(,exp...)(,)(;)
fn - Kanalnummer (1-7)
exp - Was hier angegeben wird, kann entweder ein String in Anführungszeichen, eine String-Variable oder eine numerische Variable sein.

Beispiele :
```
PRINT #3,"Hallo, da draussen",1,2,3
PRINT #OUTFILE, NAME$
```

Beschreibung :

Bei der Verwendung einer Kanalnummer mit dem PRINT-Befehl, werden die angegebenen Ausdrücke in ein Diskfile geschrieben, das vorher mit dem OPEN-Kommando geöffnet wurde.

Beachte: Die Buchstaben werden genauso in das Diskfile geschrieben, wie sie ohne Verwendung einer Kanalnummer auf den Bildschirm geschrieben würden.

Ein (,) Komma hinter dem fn leitet einen Tabulatorvorschub ein, bevor der angegebene Ausdruck gePRINTed wird; ein (;) Strichpunkt nicht, daher wird normalerweise der Strichpunkt verwendet.
In Kapitel 4.4 INPUT finden Sie weitere Informationen und ein Beispielprogramm.

4.11 PUT (Lege(ab))

Kommando : PUT

Zweck : schreibt ein einzelnes Datenbyte in das angegebene File.

Verwendung : ATARI-BASIC und BASIC XL

Anwendungsweise : PUT #fn,var
fn - Kanalnummer (1-7)
var - ein arithmetischer Ausdruck

Beispiele : PUT #3,65
PUT #OUTFILE, ASC("A")

Beschreibung:

Die PUT-Anweisung wird verwendet, um ein einzelnes Datenbyte in ein File zu schreiben. Die mit dem PUT-Befehl verwendete Kanalnummer muß mit der, des vorher für dieses File geöffneten Kanals übereinstimmen.

Beachte: Daten, die mit PUT in ein File geschrieben wurden, können gewöhnlich nur mit GET wieder ausgelesen werden. (siehe dort: Kapitel 4.3 mit Beispielprogramm)

4.12 SAVE (Speichere)

Kommando : SAVE

Zweck : speichert ein BASIC-Programm in interner (tokenisierter) Form (nicht ATASCII) auf Diskette ab.

Anwendung : ATARI-BASIC und BASIC XL

Anwendungsweise : SAVE Filename

Beispiele : SAVE"D2:GAME.SAV"
SAVE FILE$

Beschreibung :

Das SAVE-Kommando wird verwendet, um ein BASIC-Programm im internen Code auf Diskette abzuspeichern. Diese Form ist normalerweise platzsparender, als die Textform (So wie das Programm auf dem Bildschirm steht).

Alle Programme, die mit SAVE abgespeichert wurden, müssen mit LOAD oder RUN eingelesen bzw. gestartet werden.
Siehe auch bei LOAD und RUN für weitere Infomationen.

4.13 XIO (BASIC-Ein-und Ausgabe-Kommando)

Kommando : XIO

Zweck : Dies ist das universelle Ein/Ausgabe-Kommando des BASIC. Wenn das BASIC kein Kommando mehr bereitstellt, sollte man es mit XIO versuchen, hier hat man meist Erfolg.

Anwendung : ATARI-BASIC und BASIC XL, beachten Sie aber bitte die entsprechenden Hinweise bei den Unterkommandos.

Anwendungsweise : XIO,Unterkommando, #fn, aux1, aux2, Filename
Unterkommado: Bitte sehen Sie bei den im folgenden aufgeführten Unterkommando-Beschreibungen nach.

fn - Kanalnummer. Im Gegensatz zu den meisten anderen Befehlen erwartet XIO nicht, daß ein spezieller Kanal geöffnet wurde, sondern öffnet diesen, abhängig vom verwendetetn Unterkommando selbsttätig.

aux1 + aux2: meistens beinhalten diese den Wert 0, welcher unverändert an das Betriebssystem und damit die Peripherieverwalter weitergegeben wird, so daß verschiedene Peripheriegeräte verschiedene Werte erwarten können. Keines der folgenden Beispiele verwendet diese Variablen.

Beschreibung:

Obwohl das XIO-Kommando, wie bereits erwähnt, sehr universell ist, sollen hier nur die 4 meistgebrauchtesten Unterkommandos für den BASIC-Programmierer behandelt werden. Für weiterführende Informationen sei auf Kapitel 10 dieses Handbuches und speziell auf das ATARI 850 Interface-Handbuch verwiesen.

Die im folgenden behandelten Unterkommandos werden in gleicher Weise wie bisher (wie z.B. die BASIC-Anweisungen) beschrieben.

Unterkommando : RENAME

Zweck : um Diskfiles umzubenennen.

Anwendungsweise : XIO 32, #fn, 0, 0, Filename
fn - Kanalnummer eines ungeöffneten Kanals.
Filename - ein erlaubter Filename, gefolgt von einem zweiten im gleichen String, durch ein Komma getrennt. Der zweite Filename darf keine Diskettenstationspezifikation beinhalten.

Beschreibung:

Meist wird für fn (Kanalnummer) der Kanal 7, der für System-Ein-und Ausgabe (was dies hier ist) reserviert ist, verwendet. Das einzig gewöhnungsbedürftige ist die Form des Filenamen hier:

```
XIO 32, #7, 0, 0, "D:TEST.SAVE,OLDTEST.SAV"
DIM FL$(100)
INPUT FL$
FL$(LEN(FL$)+1 = ",BACKUP"
XIO 32,#7,0,0,FL$
```

Beachten Sie bitte nochmals, daß der zweite Filename keine Diskettenstationspezifikation besitzt.

Obwohl die XIO-Funktion für dieses Unterkommando auch im BASIC XL verwendet werden darf, gibt es für diese User eine einfachere Möglichkeit, indem Sie den RENAME-Befehl des BASIC XL verwenden. Mehr hierzu entnehmen Sie bitte Ihrem BASIC XL-Handbuch.

Unterkommando : ERASE

Zweck : um Diskfiles zu löschen.

Anwendungsweise : XIO 33, #fn, 0, 0, Filename
fn - Kanalnummer eines ungeöffneten Kanals.
Filename - ein erlaubter Filename; es dürfen auch wildcards verwendet werden, um mehrere Files zu löschen.

Beschreibung:

Wenn sich das angegebene File auf der Diskette befindet und nicht schreibgeschützt ist (siehe nächstes Unterkommando), so wird es unwiederbringlich von der Diskette gelöscht.

Verwenden Sie dieses Unterkommando mit Vorsicht, hauptsächlich bei der Verwendung von wildcards, d.h wenn der angegebene Filename Sternchen enthält, da hier alle Programme, bei denen eine übereinstimmung festgestellt wird, gelöscht werden.

Beispiele: XIO 33,#7,0,0,"D":OLDPROG.SAV"
löscht das einzelne File OLDPROG.SAV in LW 2.
XIO 33,#7,0,0,"D:*.BAK"
löscht alle Programme mit dem Extender .BAK von der Diskette in LW 1.

Obwohl die XIO-Funktion für dieses Unterkommando auch im BASIC XL verwendet werden darf, gibt es für diese User eine einfachere Möglichkeit, indem Sie den ERASE-Befehl des BASIC XL verwenden. Mehr hierzu entnehmen Sie bitte Ihrem BASIC XL-Handbuch.

Unterkommando : PROTECT (oder:LOCK)

Zweck : um Diskfiles zu schützen.

Anwendungsweise : XIO 35, #fn, 0, 0, Filename
fn - Kanalnummer eines ungeöffneten Kanals.
Filename - ein erlaubter Filename; es dürfen auch wildcards verwendet werden, um mehrere Files zu löschen.

Beschreibung:

Alle Programme, die mit dem angegebenen Filenamen übereinstimmen, werden geschützt. Hierbei wird einfach ein Flag im Betriebssystem für dieses File gesetzt, das dem OS verbietet irgendwelche Modifikationen an diesem File durchzuführen oder es gar zu löschen. Diese Programme kann man auch bei einer Auflistung des Directory mit einem vorangestellten Sternchen erkennen; ungeschützte Programme haben an dieser Stelle einfach ein Leerzeichen.

Beispiele: XIO 35,#7,0,0,"D:*.*"
schützt alle Files auf der Disk in LW 1.
XIO 35,#7,0,0,"D4:DOS.SYS"
schützt das File DOS.SYS auf der Disk in LW 4.

Obwohl die XIO-Funktion für dieses Unterkommando auch im BASIC XL verwendet werden darf, gibt es für diese User eine einfachere Möglichkeit, indem Sie den PROTECT-Befehl des BASIC XL verwenden. Mehr hierzu entnehmen Sie bitte Ihrem BASIC XL-Handbuch.

Unterkommando : UNPROTECT (oder:UNLOCK)

Zweck : um Diskfiles zu entschützen.

Anwendungsweise : XIO 36, #fn, 0, 0, Filename
fn - Kanalnummer eines ungeöffneten Kanals.
Filename - ein erlaubter Filename; es dürfen auch wildcards verwendet werden, um mehrere Files zu löschen.

Beschreibung:

siehe vorhergehendes Kapitel. UNPROTECT entschützt zuvor mit PROTECT geschützte Programme.

Beispiele: XIO 36,#7,0,0,"D2:*.COM"
entschützt alle Files mit dem Extender .COM auf der Disk in LW 2.
XIO 36,#7,0,0,"D1:DOS.SYS"
entschützt das File DOS.SYS in LW 1.

Obwohl die XIO-Funktion für dieses Unterkommando auch im BASIC XL verwendet werden darf, gibt es für diese User eine einfachere Möglichkeit, indem Sie den UNPROTECT-Befehl des BASIC XL verwenden. Mehr hierzu entnehmen Sie bitte Ihrem BASIC XL-Handbuch.

Kapitel 5: Die internen DOS XL Kommandos.

Als interne Kommandos bezeichnet man die, die man jederzeit beim Erscheinen des D1: (oder D2: etc) im Kommando Prozessor(CP) eingeben kann.

Diese Kommados werden durch aufrufen eines bestimmten Codes, der beim Einladen festgelegt wurde, ausgeführt, d.h. sie müßen nicht erst extra von Disk eingeladen werden wie bei den externen Kommandos. Im folgenden findet sich eine Liste der meistgebrauchtesten internen Kommandos:

```
DIRECTORY - Listet das Directory
PROTECT   - Schützen
UNPROTECT - Entschützen
ERASE     - Löschen
RENAME    - Umbenennen
LOAD      - Binär laden
SAVE      - Binär speichern
RUN       - Programm an einer bestimmten Adrese im Speicher
            ausführen.
CARTRIDGE - Zum Modul
TYPE      - Textfile auf den Bildschirm schreiben.
§         - Ein Batch-File ausführen
Dn:       - voreingestelltes LW ändern.
```

und einige andere

Alle internen Kommandos können mit ihren drei Anfangsbuchstaben abgekürzt werden, da das OS/A+ aus Platzgründen nur nach den ersten drei Buchstaben eines solchen Befehls sucht. Alle Kommandos werden in den folgenden Unterkapiteln behandelt, vorab aber, um ein Gefühl für die internen Befehle zu bekommen, ein Blick auf die Directory-Anweisung. Vorab beachten Sie bitte, daß das D1: am Bildschirm vom CP als Grundeinstellung bereitgestellt wird.

```
D1:DIRECTORY    - listet alle Files in LW1.
D1:DIRTY        -   "    "    "    "   "
D1:DIR          -   "    "    "    "   "
D1:DIR *.*      -   "    "    "    "   "
D1:DIR D1:      -   "    "    "    "   "
D1:DIR D1:*.*   -   "    "    "    "   "
D1:DIR D2:      -   "    "    "    "  LW2.
D1:DIR D2:*.*   -   "    "    "    "   "
D1:DIR *.OBJ    - alle Files mit dem Ext.OBJ in LW1.
D1:DIR D2:*.ASM -   "    "    "    "  .ASM in LW2.
```

Nun folgen detaillierte Beschreibungen der einzelnen internen Befehle in alphabetischer Reihenfolge.

5.1 @

Kommando : @

Zweck : Dieses Kommando leitet die Ausführung eines Batch-Kommandos-Files ein.

Anwendung : @Filename

Der Name eines Files, das den Extender .EXC besitzt und CP-Kommandos enthält. Es sollte nur der primäre Filename ohne Extender eingegeben werden.

Beschreibung:

Das @-Kommando beauftragt das Betriebssystem Anweisungen aus einem Batch-File zu lesen und auszuführen. Dieses File ist ein Textfile, das sowohl interne, als auch externe OS/A+ Kommandos beinhalten kann; nehmen wir z.B. an, das File TEST.EXC enthält folgende Ausdrücke:

```
DIR D:
DIR D2:
END
```

Die Eingabe des Kommandos:

```
@TEST
```

würde das Betriebssystem veranlassen, die Kommandos aus dem File TEST herauszulesen, ein Inhaltsverzeichnis der in LW 1 liegenden und danach eines der in LW 2 liegenden Diskette am Bildschirm auszugeben.

In den Kapiteln 9 und 7.6 finden sich genaue Beschreibungen zum Erstellen von eigenen Batchfiles.

Beachte: Der Extender .EXC, der ein Batch-File kennzeichnet, darf beim Aufruf mit @ nicht angegeben werden, ansonsten wird eine Fehlermeldung ausgegeben.

Ein in ein Batchfile eingebundenes CAR-Kommando unterbricht die Batch-File-Ausführung.

5.2 CAR (Cartridge-Zum Modul)

Kommando : CAR

Zweck : übergibt dem Modul oder dem BASIC das Kommando.

Anwendung : CAR
ohne weitere Angaben.

Beschreibung:

Die CAR-Anweisung ermöglicht es vom DOS XL ins BASIC zu gehen und dort zu bleiben, bis ein DOS-Kommando die Kontrolle wieder an das DOS übergibt.

Achtung: Einige Module erlauben keinen DOS-Zugriff, so daß diese nicht mit dem DOS XL zusammenarbeiten.

Sollte sich kein Modul im Slot befinden oder hat man ohne BASIC gebootet, dann erscheint eine entsprechende Fehlermeldung.

5.3 Dn:

Kommando : D1:, D2:, D3. oder D4:

Zweck : ändert die aktuelle Station je nach Angabe.

Anwendung : Dn:
nur bei mehreren Laufwerken.

Beschreibung:

Nach dem Booten des CP ist die Grundeinstellung D1: mit der man am Bildschirm zu einer Eingabe aufgefordert und daran erinnert wird, welches das aktuelle Laufwerk ist.

Jedesmal, wenn man einen Filenamen in den CP eingibt, verwendet dieser, wenn keine spezielle Station mit angegeben wurde, die Grundeinstellung. Dies ist bei einem Laufwerk sowieso klar. Besitzt man mehrere Laufwerke und gibt das Kommando "D2:" ein, so ist nun diese Station die Grundeinstellung, d.h. jedes ab jetzt eingegebene File wird auf D2: vermutet.

Beachten Sie bitte, daß das DOS XL Laufwerke von D1: bis D8: ansprechen kann, die Standard-Laufwerke aber nur von D1: bis D4: angesprochen werden können. Diskettenstationen von anderen Herstellern dagegen können möglicherweise auch mit höheren Nummern angesprochen werden.

Achtung: In der Ursprungsfassung kann DOS XL nur die Laufwerke D1: und D2: addressieren. Beachten Sie bitte Kapitel 7.5 und Anhang A, um dies zu ändern.

5.4 DIR

Kommando : DIRectory

Zweck : Auflisten des Inhaltsverzeichnisses der aktuellen Disk.

Anwendung : DIR (Dn:) (Filename) (Ausgabegerät)
optionaler Filename und optionales Ausgabegerät

Beschreibung :

Das DIR-Kommando durchsucht das Inhaltsverzeichnis der angegebenen Diskette nach allen Files, die mit dem genannten Filenamen übereinstimmen und gibt die Namen gefolgt von der Länge (in Sektoren) danach auf dem Bildschirm oder einem anderen angegebenen Gerät, aus. Ein vorangestelltes Sternchen gibt auch hier wieder an, daß ein File schreibgeschützt ist.

Der Filename kann jede erlaubte Buchstabenkombination sein und auch die wildcards dürfen verwendet werden. Die Anweisung

```
DIR D1: P:
```

würde das Directory auf dem Drucker ausgeben.

Achtung: gibt man ein Diskfile als Empfänger der Auflistung an, funktioniert dies meist nur, wenn die Zieldiskette nicht die Ausgangsdiskette ist.

5.5 END

Kommando : END

Zweck : beendet die Ausführung eines Batchfiles

Anwendung : END (ohne weitere Angaben)

Beschreibung :

Das END-Kommando veranlaßt DOS XL keine weiteren Anweisungen aus einem Batchfile herauszulesen und wieder eine Benutzereingabe zu erwarten. END hat keinen Effekt außerhalb eines Batchfiles.

5.6 ERA

Kommando : ERAse

Zweck : Files von der Diskette zu löschen.

Anwendung : ERA(Dn:)Filename

Beschreibung :

Die ERA-Anweisung löscht die angegebenen Files (fast) unwiederbringlich von der Diskette.

Vergessen Sie nicht die richtigen Laufwerksnummer einzugeben und wichtige Programme mit der PROtect-Anweisung zu schützen.

Anmerkung: Sollte das ERAse-Kommando keine übereinstimmung des angegebenen Filenamen mit den Filenamen auf der Disk feststellen können, so erscheint ein FILE NOT FOUND-Error (File nich gefunden).

5.7 LOA

Kommando : LOAd

Zweck : Diskfiles in den Speicher laden.

Anwendung : LOA(Dn:)Filename

Beschreibung:

Das LOA-Kommando erlaubt es binäre Diskfiles in den Userspeicher einzuladen. Diese Files müssen die allgemeinen Bestimmungen des normalen Betriebssystems für Binärfiles befolgen.

Jedem Segment im Speicher müßen zwei Addressen voranstehen und zwar die Start und Endaddresse im RAM-Speicher des Segments. Dem Gesamtfile müßen zwei Bytes voranstehen, in dem alle Bits gesetzt sind ($FF,$FF). Dieses Format entspricht dem, das auch die ATARI-Assembler-Cartridge und die meisten anderen Produkte in der Richtung auch verwenden (z.B.ACTION! und MAC/65 von OSS).

5.8 NOS

Kommando : NOScreen

Zweck : Schaltet das Kommando-Echo am Bildschirm bei der Batch-Verarbeitung ab.

Anwendung : NOS ohne weitere Angaben

Beschreibung:

Normalerweise werden alle gerade ausgeführten Kommandos beim Ablauf einer Batchverarbeitung synchron am Bildschirm angezeigt, als ob sie vom Benutzer eingetippt würden. Mit Hilfe des NOS-Kommandos kann dieses Echo abgeschaltet werden; dies läßt sich mit dem Kommando SCR wieder umstellen.
Dieses Kommando arbeitet nur im Zusammenhang mit der Batch-Verarbeitung.

5.9 PRO

Kommando	: PROtect
Zweck	: schützt Files vor versehentlichem Löschen oder überschreiben
Anwendung	: PRO (Dn:)Filename
Angaben	: Filename
Anmerkung	: Wildcards dürfen verwendet werden, um mehrere oder alle Files auf einer Disk zu schützen.

5.10 REM

Kommando : REMark

Zweck : schreibt Bemerkungen auf den Bildschirm während einer Batchverarbeitung.

Anwendung : REM jede beliebige Zeichenfolge.

Angaben : ein String von 0 oder mehr Zeichen.

5.11 REN

Kommando : REName

Zweck : Umbenennen eines Files.

Anwendung : REN Quellfilename Zielfilename

Angaben : zwei Filenamen.

Achtung: Bitte verwenden Sie bei REName keine wildcards, da hier sonst schnell etwas durcheinanderkommen kann.

5.12 RUN

Kommando : RUN

Zweck : übergibt die Kontrolle einer Adresse im Speicher.

Anwendung : RUN hexadezimale Adresse.

Beschreibung:

Es wird vom DS sofort ein Sprung zu der angegebenen Adresse ausgeführt, ohne vorherige Kontrolle, ob sich dort auch ein sinnvolles Programm befindet. Eine evtl. angegebene hexadezimale Zahl, muß entweder 3 oder 4 stellig sein.

Jedesmal, wenn mit dem LOAD-Kommando ein Programm in einen bestimmten Speicherbereich eingeladen, oder eine anderer externer Befehl ausgeführt wird, wird auch die Adresse im RUNLOC-Zeiger aktualisiert. Daher kann man ein Programm, das man zwischendurch z.B. zum DOS verlassen hat, einfach durch Eingabe von RUN wieder starten.

5.13 SAV

Kommando : SAVe

Zweck : speichert einen Memory-Inhalt auf Disk ab.

Anwendung : SAVE Filename Anfangsadresse Endadresse

Angaben : Filename, hexadezimale Start-und Endadresse.

Beschreibung:

Das SAV-Kommando ermöglicht es einen Teil des Speichers auf Diskette in binärer Fileform zu speichern. Die beiden angegebenen Adressen spezifizieren den abzuspeichernden Bereich; natürlich muß hier die zweite Zahl größer sein als die erste. Ein File, das geSAVed wurde kann später mit LOAD vom DOS aus wieder eingeladen werden.

Beispiel:

```
SAVE PAGE4000 4000 40FF
```

speichert die 256. Page des Speichers auf Disk unter dem Namen PAGE4000 ab.

5.14 SCR

Kommando : SCReen

Zweck : bewirkt, daß während einer Batchausführung, die gerade aktuellen Schritte am Bildschirm angezeigt werden.

Anwendung : SCR ohne weitere Angaben.

5.15 TYP

Kommando : TYPe

Zweck : schreibt ein ASCII oder ATASCII-File auf den Bildschirm oder zu einem anderen File.

Anwendung : TYP (Dn:)filename (Ausgabefile)

Angaben : Filename, der Name jedes beliebigen Textfiles, optional das Ausgabefile

Beschreibung:
Das TYPe-Kommando erlaubt es Textfiles auf den Bildschirm oder in ein anderes File zu kopieren. Ist das optionale Ausgabefile nicht angegeben, wird das File auf den Bildschirm kopiert. Um z.B. die Kommandos des STARTUP.EXC-Files auf der Masterdisk anzusehen, wäre folgendes einzugeben:

```
TYP STARTUP.EXC
```

Gibt man ein optionales Ausgabefile an, wird das Textfile entsprechend dorthin kopiert. z.B. zum Drucker:

```
TYP STARTUP.EXC P:
```

Die umgekehrte Anwendung nimmt man zum Erstellen eine solchen Textfiles:

```
TYP E: STARTUP.EXC
```

würde ein mit >CONTROL-3< abgeschlossenes vorher am Bildschirm eingegebenes File auf der Disk abspeichern; hier darf man aber keine Cursor-Kontrolltasten verwenden, da eine abgeschlossene Zeile so in den Speicher übernommen wird. Daher eignet sich dies nur für kurze Textfiles. Beachten Sie bitte, daß das Original-STARTUP.EXC überschrieben werden könnte.

Schließlich kann man TYP noch dazu verwenden, Text von einem in ein anderes Textfile zu kopieren, indem man für Ein-und Ausgabefile entprechend zwei Filanamen angibt.

5.16 UNP

Kommando : UNProtect

Zweck : File(s) entschützen.

Anwendung : UNP(Dn:)Filename

Angaben : ein Filename, wildcards dürfen verwendet werden.

Kapitel 6: Die externen DOS XL-Kommandos.

Externe Kommandos sind Programme, die vom CP gestartet werden. Jedes File mit dem Extender .COM kann von DOS XL als externes Kommando verwendet werden; das DOS XL COPY-Programm, das Sie durch Anwenden des DIRECTORY-Befehls in der Disketteninhaltsauflistung unter dem Namen COPY.COM finden, ist ein solches externes Kommando; d.h. wenn man im CP das Wort COPY eintippt, dann wird der Inhalt dieses Programms ausgeführt.

Externe Kommandos sind kein Bestandteil des eigentlichen Betriebssystems, so daß das System beim Aufruf eines solchen zuerst auf der Diskette nach einem .COM-File mit dem entsprechenden primären Namen sucht, dieses einlädt und startet. Sollte sich das gewünschte Programm nicht auf der Diskette befinden, so wird die Fehlermeldung: FILE NOT FOUND (File nicht gefunden) ausgegeben; d.h. wann immer man ein externes Kommando aufruft, muß sich die Master-Disk (oder die Arbeitskopie) in der Diskettenstation befinden.

Nach dem Eintippen einer Kommandozeile im CP, prüft dieser, ob die ersten drei Buchstaben auf einen internen Befehl hinweisen, ist das nicht der Fall, wird zunächst ein externer Befehl angenommen. Aus diesem Grunde darf aber auch ein externes Kommando nie mit den selben drei Anfangsbuchstaben wie ein interner Befehl anfangen, so daß man z.B. das File PROCESS3.COM nicht einfach durch Eintippen von PROCESS3.COM aufrufen kann, da das System hier den internen Befehl PROtect erkennen würde.
Abhilfe:

1) Den externen Befehl umbenennen.
2) Die Kommandos:
   LOAD PROCESS3.COM >RETURN<
   RUN >RETURN< eingeben.

Um ein externes Kommando auszuführen, geht DOS XL wie folgt vor:

Es wird:

1) die Diskettenstationsnummer der Grundeinstellung vorangestellt, wenn keine angegeben wurde,
2) überprüft, ob sich ein .COM Extender am File auf der Disk befindet,
3) ein Kanal zur Eingabe für dieses File geöffnet,
4) das File auf das richtige Speicherformat überprüft,
5) das File geladen und ausgeführt.

Anmerkung: a) Sollte bei einem dieser Punkte ein Fehler auftreten, so werden entsprechende Fehlermeldungen ausgegeben.
b) Schritt 1 setzt voraus, daß überhaupt eine Diskettenstationspezifikation angegeben werden darf.

Geben Sie nie den Extender .COM mit an, da das Ergebnis nach Durchlaufen der verschiedenen Tests D1:COPY.COM.COM lauten würde, was natürlich ungültig ist und eine Fehlermeldung herbeiführt.

Einige externe Kommandos befinden sich ja schon auf der DOS XL Masterdisk, wobei das System nicht auf diese paar externen Kommandos beschränkt bleibt, sondern der User sich jederzeit zusätzliche Kommandos selbst programmieren kann. Wenn Sie interessiert sind Ihre eigenen Anweisungen zu schreiben, so lesen Sie bitte Kapitel 8.

Ist ein externes Kommando schon einmal im Speicher gelaufen und man ist zwischendurch ins DOS zurückgekehrt, so kann man meist mit RUN ohne weitere Angaben wieder zu diesem Programm gelangen. Ausnahmen zu dieser Regel sind die Programme COPY, COPY24, SDCOPY und CONFIG.

Die folgenden Unterkapitel geben Beschreibungen zu den externen Kommandos, die sich als Standardausrüstung auf der DOS XL-Masterdisk befinden in alphabetischer Reihenfolge.

6.1 C65

Kommando : C65

Zweck : lädt und startet den OSS C/65-Compiler.

Anwendung : nur für Besitzer des C/65

Anwendungsweise: C65 Quellfile Zielfile(-T)

Angaben : Zwei Filenamen

Option : -T bezieht den C/65 Quelltext in den Assembler Output mit ein.

Beschreibung :

Dieses Kommando lädt und startet das File C65.COM, den kleinen C-Compiler von OSS. Zwei Filenamen werden benötigt: der erste muß ein Textfile mit C/65-Quelltext und die zugehörigen Angaben beinhalten; unter dem zweiten wird ein MAC/65-kompatibles Machinenspracheprogramm geschrieben (oder überschrieben, falls bereits ein File mit diesm Namen existierte).

Option:
Wird die "-T"-Option mit angegeben, enthält das erzeugte MAC/65-File die vom User eingegebenen C/65-Quelltext-Zeilen; d.h. jedem Assembler-Code steht der entsprechende C/65-Quelltet voran.

** Weitere Informationen befinden sich im C/65-Handbuch. **

** Das Programm MAC/65 ist nicht im Lieferumfang des DOS XL enthalten. **

6.2 CLRDSK

Kommando : CLRDSK

Zweck : Formatieren einer Diskette im 810-Format.

Anwendung : für nicht-ATARI Diskettenstationen.

Anwendungsweise: CLRDSK

Angaben : keine

Optionen : keine

Beschreibung :

Dieses Hilfsprogramm zwingt eine Atari-fremde Diskettenstation eine Diskette im 810-Format zu formatieren. Sollte ein Programm nicht mit einer Diskette zusammenarbeiten, die in Ihrem atari-fremden Laufwerk initialisiert wurde, so wird sie es sicher nachdem man sie mit CLRDSK formatiert hat.

Anmerkung: CLRDSK formatiert zuerst die Diskette und schreibt dann Nullen in alle Sektoren außer denen der Directory, der Boot- und VTOC-Sektoren.

Verwenden Sie CLRDSK generell nicht, wenn Sie nicht extra vom Hersteller Ihrer Fremd-Diskettenstation dazu aufgefordert werden. Sollte sich das Programm CLRDSK nicht auf der Master-Disk Ihrer Station befinden, so ist es zu deren Gebrauch nicht notwendig.

6.3 CONFIG

Kommando : CONFIG

Zweck : ermöglicht das Ändern des Status eines veränderbaren Laufwerks.

Anwendung : für Benutzer von konfigurierbaren Diskettenstationen.

Anwendungsweise: CONFIG (param1,param2...)(-N)

Angaben : eine entsprechende Liste von Parametern, die die entsprechenden Konfigurierungen durchführen.

Optionen : -N es wird keine Konfigurationstabelle ausgegeben.

Beschreibung :
Wenn keine Parameter angegeben sind, berichtet diese Funktion nur vom gegenwärtigen Status aller angeschlossenen Diskettenstationen.
Sind ein oder mehrere Parameter angegeben, werden diese als Informationen für die Stationen, die konfigurierbar sind behandelt. Ein solcher Parameter besteht aus einer Zahl (zwischen 1 und 8) gefolgt von einer ein-oder zweistelligen Buchstabenfolge (dem Modus). Die Zahl bezeichnet die Nummer des gemeinten Laufwerkes (D1:bisD8:), die Buchstabenfolgen bezeichnen verschiedene Modi, wie in der folgenden Tabelle zu sehen ist:

Modus	Bedeutung
S	Einstellung für eine Station als einseitig in einfacher Schreibdichte.
D	Einstellung für eine Station als einseitig in doppelter Schreibdichte.
DD	Einstellung für eine Station als doppelseitig in doppelter Schreibdichte.

Optionen:
Normalerweise wird die Konfigurationstabelle der entsprechenden Diskettenstation aufgelistet, durch die Option -N kann dies jedoch unterdrückt werden.

Anmerkung: DOS XL, Version 2 kann auf die zweite Seite einer doppelseitigen Diskette in doppelter Schreibdichte nicht zugreifen. Verwenden Sie DOS XL, Version 4, wenn Sie ein solches Laufwerk besitzen.

Beispiel :

CONFIG 1D 2DD

veranlaßt, daß D1: als einseitiges Laufwerk in doppelter Dichte behandelt wird und D2: doppelseitig in doppelter Schreibdichte.

Anmerkungen:
Wird eine System-Konfiguration durchgeführt, so wird der Filemanager neu initialisert und der Systemstatus berichtigt, als ob die Anweisung CONFIG ohne Parameter gegeben worden wäre.

Falls eine Konfigurations-Anweisung ungültig ist, d.h. ist ein Laufwerk nicht via Software umstellbar, so erscheint eine entsprechende Meldung.

6.4 COPY

Kommando : COPY

Zweck : zum Kopieren von Files.

Anwendung : COPY Quellfile Zielfile (-FQSW) oder
COPY Filename (-FQSW)

Angaben : ein oder zwei Filenamen

Optionen : -F bereits bestehendes File wird überschrieben.
-Q spezielle Nachfrage für jedes zu kopierende File.
-S Kopieren auf der gleichen Diskette.
-W Auf die Userantwort warten, bevor mit dem Kopieren angefangen wird.

Beschreibung :
Mit dem COPY-Kommando werden ein oder mehrere Files kopiert, ohne das Quellfile zu verändern. In der oben zuerst genannten Anwendung, werden alle Files, die mit dem angegebenen Quellfilenamen übereinstimmen, unter dem/den neuen Zielfilenamen kopiert, das/die auf der gleichen oder auf einer anderen Diskette erzeugt werden kann/können. In der zweiten Anwendung, würden alle Files, die mit dem Quellfilenamen übereinstimmen unter dem gleichen Namen auf die gleiche Diskette kopiert werden. Dies ermöglicht das Kopieren in einem System mit einer Diskettenstation. Die Quell-und Zielspezifizierungen sollten in einer der folgenden Formen gehalten sein:

1) (Dn:)Filename
2) Dn:

Bei Nummer 1) ist die Diskettennummer optional, da die momentan aktuelle Grundeinstellung angenommen wird, wenn hier nichts angegeben wird.
Bei Nummer 2) würden alle Files von oder auf eine Diskette kopiert werden.

Optionen: Die "-F"-Option veranlaßt, daß ein bereits unter dem Zielfilenamen bestehendes Programm überschrieben wird; wurde diese Option nicht ausgewähl~~t wer~~den solche Files nicht überschrieben.

Bei Verwendung der "-Q"-Option wird man vor dem Kopieren jedes einzelnen Files gefragt, ob es kopiert werden soll.

Durch die "-S"-Funktion teilt man dem COPY-Programm mit, daß das Kopieren auf einer Diskettenstation vor sich gehen soll; hierbei wird man jeweils aufgefordert die Quell-bzw.Zieldiskette einzulegen.

Option "-W" läßt das Programm so lange warten, bis der Benutzer die entsprechende Diskette eingelegt hat, bevor der Kopiervorgang beginnt.

Achtung: Bitte verwenden Sie das COPY-Programm nicht zum Kopieren einer in einfacher Schreibdichte auf eine in doppelter Schreibdichte formatierten Diskette. Verwenden Sie hierzu bitte das Hilfsprogramm SDCOPY und beachten Sie Kapitel 6.13.

Beispiele:

COPY *.* - kopiert alle Files in Laufwerk 1 auf eine andere Diskette in der gleichen Station. Man wird jeweils auf das nötige Austauschen der Disketten aufmerksam gemacht; allerdings ist das File DUPDSK ein praktischeres Hilfsmittel, um diese Funktion zu verarbeiten.

COPY *.COM D3: -F - kopiert alle Files von der aktuell eingestellten Diskettenstation auf die Diskette in LW 3 (Vorsicht:Dies könnte die voreingestellte sein !), wobei alle unter gleichem Namen bereits bestehende gelöscht und überschrieben werden.

COPY D2:C*.* D1: -Q - kopiert alle Files mit dem Anfangsbuchstaben "C" von der Diskette in LW2 auf die Diskette in LW1, wobei bei jedem File einzeln gefragt wird, ob es kopiert werden soll.

COPY D1:TEST D2:NEWTEST - kopiert das File TEST von Station 1 nach Station 2 unter dem Namen NEWFILE.

COPY D:TEST D1:NEWTEST -S - schreibt das File TEST auf der gleichen Diskette unter neuem Namen.

6.5 DO

Kommando : DO

Zweck : ermöglicht es mehrere Befehle in einer Kommandozeile auszuführen.

Anwendung : DO Kommando (;Kommando;Kommando...) oder
DO

Angaben : optional; Liste von Kommandos, die jeweils durch einen Strichpunkt voneinander getrennt sind.

Optionen : Keine

Beschreibung :
Mehrere Kommandos können in eine Zeile geschrieben und ausgeführt werden, wobei die Kommandos nicht auf DOS XL's interne und externe Befehle beschränkt sind. Das folgende Beispiel würde zum Beispiel die Assembler/Editor-Cartridge oder den MAC/65 aufrufen, ein Quellprogramm einladen und dieses assemblieren:

```
DO CAR;ENTER"D:PROGRAM.ASM";ASM ,#P:,#D:OBJECT
```

In der zweiten Anwendungsform fragt das DO-Hilfsprogramm den Benutzer nacheinander nach einer Reihe von gewünschten Befehlen und merkt sich diese. Gibt man bei der Frage nach einem Befehl ein >RETURN< ein, so wird die ganze Liste von Kommandos ausgeführt.

Das DO-Kommando kann auch benutzt werden, um ein BASIC-Programm (ähnlich der Funktion eines AUTORUN.SYS-Files bei ATARI-DOS) beim Booten einzuladen und zu starten, indem man die entsprechende DO-Anweisung in ein STARTUP-.EXC einbindet (Siehe Kapitel 9: Batch-Verarbeitung). Das folgende Beispiel zeigt ein STARTUP.EXC-File, das das BASIC-Programm "MENU" beim Booten startet:

```
DO CAR;RUN"D1:MENU"
```

Anmerkung: Einige Programme, die die DOS XL-Kommandozeile selbst überprüfen können ~~nicht~~ zusammen mit dem DO-Kommando angewendet werden. Die COPY-Anweisung des CP wäre ein solches Programm.

6.6 DUPDBL

Kommando : DUPDBL

Zweck : ermöglicht schnelles Kopieren von Disketten in doppelter Schreibdichte.

Anwendung : nur für Systeme mit Laufwerken, die unter doppelter Schreibdichte arbeiten.

Aufruf : DUPDBL(ohne weitere Angaben)

Optionen : Keine

Beschreibung :

Das DUPDBL-Hilfsprogramm fordert während der Ausführung die Quell-oder die Zieldiskette an und fragt, ob die Zieldisk formatiert werden soll oder nicht. Die gesamte Quelldiskette wird hier schneller als mit der COPY-Funktion kopiert. Auf jeden Fall müßen beide Disketten unter DOS XL (Version 2) im OS/A+-Betriebssystem oder ATARI-DOS 2.x in doppelter Schreibdichte formatiert sein. Ist die Ziel-mit der Quelldiskettenstation identisch, wird man während des Kopiervorgangs entsprechend aufgefordert, die Quell-bzw.die Zieldiskette einzulegen.

Siehe auch Kapitel 7.3 und 7.4

6.7 DUPDSK

Kommando : DUPDSK

Zweck : kopiert ganze Disketten in gleicher Schreibdichte.

Anwendung : für alle, außer Laufwerke in doppelter Schreibdichte.

Aufruf : DUPDSK

Angaben : keine

Optionen : keine

Beschreibung :

Das DUPDSK-Hilfsprogramm fragt während der Ausführung jeweils nach der Quell-bzw.Zieldiskette und, ob die Zieldiskette formatiert werden soll. Auch hier wird schneller kopiert als dies mit dem COPY-Befehl möglich ist. Ist die Ziel-mit der Quelldiskettenstation identisch, wird man bei Bedarf aufgefordert die entsprechende Disk einzulegen.

Achtung: Verwenden Sie DUPDSK unter DOS XL Version 2 nicht zum Kopieren von Disketten in doppelter Schreibdichte, da sonst ein heilloses Durcheinander entsteht, sondern nehmen Sie zu diesem Zweck das bereits in einem früheren Kapitel beschriebene DUPDBL.

In Kapitel 7.3 und 7.4 finden sich Informationen über das Kopieren in verschiedenen Schreibdichten.

6.8 INIT

Kommando : INIT

Zweck : formatiert Disketten, so daß diese schreib-und lesefähig werden.

Anwendung : INIT

Angaben : keine

Optionen : keine

Beschreibung:

Bei der Version 2 des DOS XL wird der Benutzer genau nach der Art der Formatierung gefragt (z.B mit oder ohne Systemfile etc...). Nachdem der Initialisierungsvorgang beendet ist, kann die Diskette zum Speichern von Daten verwendet werden.

Das INIT-Hilfsprogramm stellt folgende Optionen zur Auswahl:

1. Format Disk only (nur formatieren)
2. Format Disk and write DOS.SYS (formatieren und DOS.SYS schreiben)
3. Write DOS.SYS only (nur DOS.SYS schreiben)
4. Exit to DOS XL (zurück zum DOS)

Achtung: Wenn Sie DOS XL entweder mit der Super-Cartridge oder einem ATARI-Computer der XL-Serie verwenden, dürfen die Optionen 2 und 3 nicht dazu verwendet werden das File DOS.SYS auf die Disk zu schreiben. Um eine bootbare DOS XL-Diskette zu erstellen, sollten statt dessen die Files DOS.SYS und DOSXL.SYS mit dem COPY-Befehl auf eine vorher formatierte Diskete kopiert werden. (Eine Diskette, die nur DOS.SYS enthält, würde zwar gebootet werden, jedoch wird dann kein Gebrauch vom, in dieser Konfiguration zur Verfügung stehenden Extra-Speicher gemacht). Wurde das System mit einer Masterdisk, die kein DOSXL.SYS-File enthält gebootet, dann können die Optionen 2 und 3 ruhig verwendet werden. (Siehe auch Kapitel 3.7 und 7.7).

6.9 INITDBL

Kommando : INITDBL

Zweck : formatiert in doppelter Schreibdichte.

Anwendung : für Benutzer eines einzigen nicht-Atari-Diskettenlaufwerkes zusammen mit DOS XL.

Angaben : keine

Optionen : keine

Beschreibung:

Beachte: Dieses Hilfsprogramm ist für Benutzer, die mehr als ein LW besitzen unnötig. Verwenden Sie bitte stattdessen das Standard INIT-Kommando (Kapitel 6.8).

Mit dem INITDBL-Befehl wird in einem System mit nur einem Laufwerk eine Diskette in doppelter Dichte formatiert und mit DOS.SYS beschrieben.

Um dieses Hilfsmittel anzuwenden, booten Sie bitte die Master-Disk, tippen das INITDBL-Kommando ein und beantworten die Frage mit der Nummer 1. Bevor Sie jedoch >RETURN< drücken, entnehmen Sie bitte die Masterdisk und legen die zu formatierende ein. Ist der Initialisierungsvorgang beendet, befindet sich die Station trotzdem noch in der Konfigurierung für einfache Schreibdichte. Um ein Inhaltsverzeichnis einer Diskette in doppelter Schreibdichte zu erhalten, wenden Sie den CONFIGure (Konfiguriere)-Befehl an, um die Diskettenstation entsprechend auf doppelte Schreibdichte umzustellen und geben DIR ein.

6.10 MAC65

Kommando : MAC65

Zweck : lädt und startet den MAC/65 Macroassembler.

Anwendung : für Besitzer des MAC/65.

Aufruf : MAC65(file1 (file2 (file3)) (-A) (-D))

Angaben : optionale Angabe von ein bis drei Filenamen, die den Quell-,Listing und Objektfiles einer MAC/65-Assemblierung entsprechen.

Optionen : -A Quellfile ist in ASCII-Fotmat.
-D Assemblierung muß von Disk zu Disk verlaufen.

Beschreibung:

Diese Anweisung lädt und startet den MAC/65-Macroassembler; sind keine Filenamen angegeben, dann wird der Assembler in seinem interaktiven Modus gestartet (Editor), in dem Programme und Texte editiert oder assembliert werden können. Weitere Informationen im MAC/65-Handbuch.

Sind ein oder mehrere Files angegeben, wird der MAC/65 in seinem Batch-Modus gestartet, d.h. es wird eine einzelne Assemblierung durchgeführt und danach zum OS/A+ zurückgekehrt. Generell bewirkt diese Kommandozeile das gleiche wie der ASM-Befehl des MAC/65-Editors. Ist nur ein Filename genannt worden, so wird dieser als Quellfile vermutet, das Listing wird am Bildschirm ausgegeben und der Objekt-Code im Speicher abgelegt (aber nur, wenn dies mit der .OPT OBJ-Direktive gewünscht wurde). Ist ein zweiter Filename angegeben, wird dieser als Listing-File verwendet und nur wenn alle drei Filenamen gegeben sind, landet der Objekt-Code in einem Diskfile.

Beachte: Benötigt eine Asemblierung kein Listing aber ein Objekt-File, so kann E: als Listingfile (praktisch als Platzhalter) spezifiziert werden.

Optionen:
Durch die "-A"-Option wird mitgeteilt, daß es sich bei dem Quellfile nicht um ein Standard MAC/65-File sondern um ein File im ASCII- (oder ATASCII-) Format handelt, was der Verwendung des interaktiven Editors des MAC/65 entspricht, wobei die Kommandozeilen "ENTER#D..." und "ASM..." zur Anwendung kommen.

Die Option "-D" gibt an, daß die Assemblierung von Disk zu Disk erfolgen soll; ist diese Option nicht gewählt, so wird das Quellfile vor der Assemblierung geLOADed (oder geENTERed) und diese dann mit dem Quellfile im Speicher durchgeführt (was generell schneller ist). Sollte das Quellfile zu groß sein, um im Speicher assembliert werden zu können, kann diese Option verwendet werden, um auch größere Programme verarbeiten zu können. Beachten Sie bitte, daß auch wenn das Quellfile in den Speicher paßt, hier auch noch die Macro-und Symbol-Tabellen untergebracht werden müssen.

Anmerkung: Die "-D"-Option kann nicht zusammen mit Option "-A" verwendet werden, da das Quellfile bei Option -D ein geSAVEtes (tokenisiertes) File sein muß.

** Weitere Informationen befinden sich im C/65-Handbuch. **

** Das Programm MAC/65 ist nicht im Lieferumfang des DOS XL enthalten. **

6.11 MENU

Kommando : MENU

Zweck : Aufruf einer Liste der System-Kommandos als Hilfe für Anfänger.

Anwendung : Menu

Angaben : keine

Optionen : keine

Beschreibung:

Obwohl der CP praktischer in der Anwendung ist, wurde hier eine Möglichkeit für den Neuling geschaffen, um gut ins System hereinzukommen. Hier werden die meistgebrauchten Befehle aufgelistet.

Tippen Sie also einfach MENU gefolgt von >RETURN< ein, sobald die Dn:-Meldung des CP erscheint (Beschreibung des DOS XL-Menü in Kapitel 3).

6.12 RS232

Kommando : RS232

Zweck : installiert die seriellen Geräteverwalter ("Rn:") zum Gebrauch mit dem ATARI 850-Interface-Modul.

Anwendung : für Benutzer des 850.

Angaben : keine

Optionen : keine

Beschreibung:

Die Anweisung RS232 des DOS XL funktioniert genauso wie ein AUTORUN.SYS-File zur Initialisierung der seriellen Schnittstelle unter ASTARI-DOS2.x. Der Driver für die verschiedenen RS232-Funktionen wird ins LOMEM geladen und der R:-Geräteverwalter wird in die Tabelle der ansprechbaren Geräte aufgenommen.

Nach Eingabe der Anweisung RS232, wenn die Dn:-Meldung wieder unterhalb von RS232 erscheint, hat das Interface-Modul seine Software eingeladen.

Sollte der Bildschirm gelöscht werden und das Dn: erscheint oben am Bildschirm, ist irgendetwas während des Ladevorgangs falsch gelaufen. Höchstwahrscheinlich liefert die Software im Modul keine brauchbaren Fehlermeldungen zurück und führt stattdessen einen System-Warmstart aus (deshalb auch der gelöschte Bildschirm).

Achtung: Aufgrund eines Fehlers in der Software des 850-Interface-Moduls, bewirkt das Drücken von RESET die Auslöschung des LOMEM-Zeigers, wodurch der Speicherbereich, den der RS232-Verwalter einnimmt, ignoriert wird. (In Anhang B findet sich ein möglicher Weg dies zu beheben).

Manchmal ist das 850-Interface Modul auch viel zu intelligent für sich selbst, genaüer gesagt, um die Software neu einzuladen, muß das Modul aus-und wieder eingeschaltet werden.

6.13 SDCOPY

Kommando	: SDCOPY
Zweck	: kopiert Files in einfacher auf Files in doppelter Schreibdichte.
Anwendung	: nur für ATARI-User, die OS/A+ Version 2 benutzen. SDCOPY Quellfile Zielfile (-FQRV)
Angaben	: ein oder zwei Filenamen
Optionen	: -F = überschreibt ein evtl. schon best. File. -Q = Nachfrage vor jedem Filetransfer. -R = Kopierweise umkehren. -V = zusätzliche

Beschreibung:

Das Hilfsprogramm SDCOPY ist für DOS XL-Anwender mit ATARI-fremden Diskettenstationen gedacht. Der Zweck ist es, ein File in einfacher Schreibdichte in ein File in doppelter Schreibdichte (oder umgekehrt mit Hilfe der "-R"-Option) auf ein und derselben Station zu erstellen.

Dies funkioniert nur mit einer Station, möchten Sie mehrere Drives verwenden, so sind die Laufwerke mit CONFIG entsprechend einzustellen und der COPY-Befehl zu verwenden.

Kapitel 7: Mehrere Laufwerke, mehrere Schreibdichten.

Vieles aber nicht alles, was Sie bis jetzt gelesen haben, setzt voraus, daß man ein System mit nur einer Diskettenstation besitzt. DOS XL funktioniert natürlich auch hervorragend mit nur einem Laufwerk, zeigt aber seine wahren Fähigkeiten erst im Betrieb mit zwei oder mehr Stationen.

Haben Sie nur 810-Diskettenlaufwerke muß nicht viel über die Ausnutzung aller Vorteile gesagt werden. Generell werden Sie feststellen, daß z.B. ein Kopiervorgang mit der COPY-Funktion von einer Quelldiskette in LW1 auf eine Zieldiskette in LW 2 viel schneller vor sich geht als mit nur einer Station. Es interessiert dabei nicht welches die Quell-und die Zielstation sind, beachten Sie aber das einige Kommandos voraussetzen, daß sich die DOS XL-Masterdisk in Laufwerk 1 oder dem Laufwerk, das der Grundeinstellung (siehe Kapitel 5.3, wenn Sie nicht das Menü verwenden) entspricht, befindet.

Dies ist natürlich nicht zwingend, da die Kommandos sowieso darauf warten, daß Sie die Masterdisk dann eben nachträglich noch einlegen, bis Sie Ihre Arbeit ausführen können oder sogar eine solche Wartezeit ausgewählt werden kann (siehe Option "-W" des COPY-Kommandos). Beachten Sie auch, daß diese Warte-Optionen bei der Verwendung des Menüs bereits vorgegeben sind.

Besitzen Sie ein Atari-fremdes Laufwerk, so sollten Sie den Rest dieses Kapitels unbedingt lesen.

Viele der in den folgenden Unterkapitel beschriebenen Funktionen, sind nicht so ohne weiteres vom normalen DOS XL-Menü erreichbar. Man hat nun zwei Möglichkeiten: Entweder man verwendet das "X"-Kommando im Menü, bei dem man nach einer Kommandozeile gefragt wird, die dann an den CP weitergegeben wird, oder man verwendet gleich den Kommando Prozessor(CP). Hier ist auch wieder nur einfach das Kommando einzugeben, wie im folgenden jeweils gezeigt wird.

Beachten Sie bitte, daß sich bei Auswahl der Kommandos CONFIG, INITDBL, INIT, COPY und SDCOPY die Masterdiskette in LW 1 befinden muß; ansonsten erscheint eine ~~FILE~~ NOT FOUND-Meldung.

7.1 Einstellung für mehrere Laufwerke

DOS XL Version 2 ist in der Lage jede beliebige Kombination von bis zu 8 Laufwerken in einfacher oder doppelter Schreibdichte anzusprechen. Besitzen Sie eine Station, welche doppelte Schreibdichte verarbeitet und eine oder mehrere, die nur einfache Schreibdichte vertragen, so sollten Sie die Station mit der doppelten als LW 1 verwenden (Bitte sehen Sie im Handbuch des Herstellers für diese Einstellung nach), da man dann DOS XL, sowohl in einfacher als auch in doppelter Dichte booten kann.

Ist ein Laufwerk fähig beide Schreibdichten zu verarbeiten, kann man generell abschätzen in welchem Modus es sich beim Booten befindet: Handelt es sich um LW 1, so wird es sich der Schreibdichte der eingelegten Masterdisk anpassen. Ist es ein anderes, so holt es sich die entsprechenden Angaben aus dem Disk-Controller (siehe auch hier wieder das Handbuch des Herstellers).

Die DOS XL-Masterdisk verarbeitet in der Grundeinstellung bis zu drei gleichzeitig geöffnete Files (das bedeutet soviel wie 3 BASIC-"OPEN"-Anweisungen ohne ein CLOSE dazwischen) in doppelter Schreibdichte auf zwei Laufwerken. Besitzen Sie mehr als zwei Laufwerke oder möchten Sie mehrere Files gleichzeitig öffnen, müssen die Werte in verschiedenen Systemvariablen geändert werden (siehe Kapitel 7.5).

Das DOS XL fragt automatisch beim Bootvorgang jede angeschlossene Diskettenstation nach deren Einstellung ab.

Ist dies geschehen, muß man, um dies zu ändern im CP den Befehl CONFIG aufrufen, der einige Optionen bereithält (siehe Kapitel 6.3). Zuerst werden drei Fähigkeiten behandelt, die man unbedingt beherrschen muß, bevor es weitergeht. Die folgenden Kommandos müssen im CP oder aus dem Menü mit dem Xtended Command-Befehl gegeben werden.

1. Möchten Sie herausfinden in welcher Schreibdichte jedes einzelne LW angesprochen werden kann, so geben Sie bitte folgendes ein:

   CONFIG >RETURN<

   Aus der am Bildschirm ausgedruckten Tabelle, kann man dann die erforderlichen Informationen auslesen: in welcher Schreibdichte sich die jeweilige Station befindet und wieviele Diskseiten DOS XL adressieren kann (bei Version 2 immer nur eine Seite). Ist eine Station nicht unter doppelter Schreibdichte anzusprechen, erscheint die Meldung: "can't configure", d.h. es läßt sich nicht feststellen. Es wird über bis zu 8 angeschlossene Diskettenstationen berichtet und man bekommt sogar gesagt welche Stationen man nicht hat.
   Achtung: Eine hier erscheinende Meldung bedeutet nicht gleichzeitig, daß das DOS auch auf dieses LW zugreifen kann (siehe Kapitel 7.5).

2. Möchten Sie eine Station von einfacher in doppelte Dichte umschalten: dann
   CONFIG 1D >RETURN< oder:
   CONFIG 2D >RETURN< etc. eingeben

   Die Zahl gibt hier die LW-Nummer und das D doppelte Dichte an. Auf diese Weise kann jede konfigurierbare Station umgeschaltet werden, wobei die komplette Tabelle angezeigt wird.

3. Soll eine Station von doppelter in einfache Dichte umgeschaltet werden so gehen Sie bitte wie oben vor, geben aber statt dem "D" ein "S" ein.

Achtung: Wann immer Sie Disketten duplizieren (entweder mit dem "D"-Kommando aus dem Menü, mit DUPDSK oder DUPDBL), vergewissern Sie sich, daß sich beide Laufwerke in der gleichen Einstellung bzgl. der Schreibdichte befinden (In den Kapiteln 7.3 und 7.4 steht mehr über das Kopieren über verschiedene Dichten hinweg).

7.2 Initialisieren anderer Schreibdichten.

In Kapitel 3.7 wurde zum Initialisieren einer Diskette im gleichen Format wie die Masterdisk das "I"-Kommando angegeben, verwendet man allerdings auch die externen Kommandos von DOS XL, bieten sich hier auch andere Lösungen an. Außerdem gibt es auch eine Ausnahme für das in 3.7 beschriebene.

1. Auch wenn Sie ein System mit nur einem LW haben, können Sie darauf eine Disk in doppelter Dichte initialisieren, obwohl diese mit einer Masterdisk in einfacher gebootet wurde. Tippen Sie hierzu:

   INITDBL >RETURN<

   ein. Dieses Hilfsprogramm fragt nur nach dem anzusprechenden Drive (in diesem Falle 1), konfiguriert das Laufwerk entsprechend auf doppelte Dichte um, formatiert die Diskette, beschreibt diese mit dem File DOS.SYS und schaltet die Diskettenstation wieder auf einfache Dichte zurück. Verwenden Sie dieses Kommando aber nicht, wenn die Masterdisk bereits in doppelter Dichte gebootet wurde, sondern statt dessen INIT oder "I" (aus dem Menü).

2. Haben Sie mehrere Laufwerke, verwenden Sie bitte CONFIG, um LW 2 oder LW 3 (oder jedes andere) entsprechend umzuschalten und wenden dannach INIT an, das die gleichen Optionen wie die "I"-Anweisung im Menü bereitstellt.

7.3 Das Kopieren zwischen verschiedenen Schreibdichten(für ein LW)

Dieses Kapitel ist nur für diejenigen gedacht, die nur ein Laufwerk besitzen, dieses sollte aber konfigurierbar (einfache/doppelte Schreibdichte) sein. Haben Sie zwei oder mehr Stationen, dann können Sie gleich zum nächsten Kapitel übergehen.

Wenn Sie ein oder mehrere Files von einer Diskette in einfacher auf eine in doppelter Dichte kopieren wollen, oder umgekehrt, brauchen Sie zuerst eine im Zielformat initialisierte Disk. Sollten Sie noch keine solche Diskette vorbereitet haben, booten Sie bitte die Masterdisk in der gewünschten Schreibdichte und verwenden die Funktion "I" (oder sehen Sie in Kapitel 7.2 für die Verwendung von INITDBL nach.

Geben Sie jetzt bitte nach Auswahl der "Q"-Funktion im Menü:

```
       SDCOPY *.* -Q  >RETURN<
oder:  SDCOPY *.* -QR >RETURN<
```

ein, wobei das erste Beispiel alle Files von einfacher in doppelte Schreibdichte kopiert und das zweite genau umgekehrt.

SDCOPY wird daraufhin eingeladen und man wird aufgefordert die entsprechende Quelldiskette einzulegen; danach wird die Directory der Diskette gelesen und man kann die Programme, die man kopiert haben möchte durch Eingabe von "Y" oder "N" einzeln auswählen oder auch nicht. Da hier nur mit einem Laufwerk kopiert wird, müssen die Disketten pro File mindestens einmal (bei längeren Files 2-bis 3-mal) gewechselt werden.

Bemerkung:

(1) Kapitel 6.13 beschreibt andere Optionen im Umgang mit SDCOPY;

(2) Entfernen Sie alle Module aus dem Slot (oder booten Sie ohne BASIC bei XL-Modellen), wenn Sie lange Files kopieren, da die Disketten dann nicht so oft gewechselt werden müssen (dies gilt nicht für die OSS SuperCartridge, die ihren Speicherplatz dem DOS XL überläßt).

7.4 Kopieren mit mehreren Diskettenststionen.

Besitzen Sie 2 Laufwerke, so verwenden Sie besser das Standard-"Copy Files"-Kommando (Option "C" im Menü), um Files von einfacher in doppelte Schreibdichte umzuwandeln.

Zuerst sollten Sie sich natürlich vergewissern, ob die Laufwerke entsprechend konfiguriert sind.

Am besten bootet man die Master-Disk in doppelter Dichte auf LW1 und verwendet den Befehl "CONFIG 2S" (siehe Kapitel 7.1), um die zweite Station auf einfache Schreibdichte zu schalten (natürlich nicht nötig, wenn es sich bei LW2 um eine ATARI 810 handelt, da diese sowieso nur in einfacher Dichte arbeitet).

Nachdem das Menü wieder erschienen ist, wählen Sie bitte "COPY" oder "C"; auf die Frage nach dem "from" (Quell-) und "to" (Ziel-) File (oder bei der Angabe in der Kommandozeile des CP), spezifizieren Sie bitte "D1:"und "D2:" entsprechend (Macht man hier einen Fehler, so wird das angegebene File nicht gefunden, macht aber nichts, einfach noch einmal mit den richtigen LW-Bezeichnungen versuchen).

Bemerkung: Das COPY-Kommando des CP ist extrem flexibel und hilfreich. Es ist bestimmt vorteilhaft, wenn man sich wenigstens diesen Befehl des CP genauer ansieht, auch wenn man sonst nicht mit dem Kommando-Prozessor arbeitet (Beachten Sie bitte, daß der "C"-Aufruf im Menü in der Grundeinstellung mit der "-Q" und "-W" -Option läuft).

Achtung: Die Menü-Option "D" (Dupliziere Disk) darf nicht für das Kopieren von einfacher in doppelte Dichte oder umgekehrt verwendet werden, da sonst sehr seltsame Dinge geschehen würden. Dementsprechend dürfen hier die CP-Kommandos DUPDSK und DUPDBL nicht für diesen Zweck gebraucht werden.

7.5 Drei oder mehr Laufwerke

Die DOS XL-Masterdisk ist in ihrer Grundeinstellung zur Zusammenarbeit mit einer oder zwei Diskettenstationen mit bis zu 3 simultan geöffneten Files in doppelter Dichte (das bedeutet soviel wie drei OPEN-Befehle im BASIC ohne eine dazwischenliegenden CLOSE-Befehl) ausgelegt. Besitzen Sie drei oder mehr Stationen, oder wollen Sie mehr als 3 Files gleichzeitig öffnen, müssen die Werte in der Systemvariablen DRVBYT und SABYTE wie folgt geändert werden.

Ändern der Anzahl der Laufwerke: die Systemvariable DRVBYT (Speicherstelle 1802 dezimal) beinhaltet in der Grundeinstellung den Wert 3 und erlaubt damit 1 oder 2 Laufwerke anzusprechen. In der untenstehenden Tabelle finden sich die Werte für die entsprechenden änderungen:

Wert:	Laufwerksnummern:
1	nur eines
3	1 oder 2
7	1,2 oder 3
15	1 bis 4
31	1 bis 5
63	1 bis 6
127	1 bis 7
255	bis zu 8

Um den Wert in DRVBYT zu ändern, gehen Sie bitte wie folgt vor:

1) Stecken Sie das ATARI-BASIC-Modul ein (nicht BASIC XL) (nur für 400/800er-Serie).
2) Vergewissern Sie sich, daß das DOS XL-Menü nicht aktiv ist, indem Sie das "Quit to DOS XL"-Kommando anwählen.
3) Gehen Sie mit "CAR" ins BASIC.
4) POKEn Sie den gewünschten Wert in die Speicherzelle 1802.
5) Drücken Sie >SYSTEM RESET<.
6) Mit "DOS" zurück ins DOS XL gehen.
7) "MENU" eingeben, um ins DOS XL-Menü zurückzugelangen.
8) Verwenden Sie das "Initialize Disk"-Kommando mit Option 3, um das DOS.SYS neu auf die Masterdisk zu schreiben.

Ändern der Zahl der gleichzeitig geöffneten Files:
Mit der Grundeinstellung der DOS XL-Diskette lassen sich bis zu 6 Files in einfacher und 3 in doppelter Dichte gleichzeitig öffnen. Sollte dies nicht ausreichen, muß der Wert der Systemvariablen SABYTE, Speicherstelle 1801 dezimal, entsprechend der folgenden Tabelle geändert werden.

Wert:	geöffnete Files in einfacher Dichte:	geöffnete Files in doppelter Dichte:
2	2	1
4	4	2
6	6	3
8	-	4
10	-	5
12	-	6

Folgende Schritte sind auszuführen, um den Wert in SABYTE zu verändern:

1) Stecken Sie das ATARI-BASIC-Modul ein (nicht BASIC XL) (nur für 400/800er-Serie).
2) Vergewissern Sie sich, daß das DOS XL-Menü nicht aktiv ist, in dem Sie das "Quit to DOS XL"-Kommando anwählen.
3) Gehen Sie mit "CAR" ins BASIC.
4) POKEn Sie den gewünschten Wert in die Speicherzelle 1801.
5) Drücken Sie >SYSTEM RESET<.
6) Mit "DOS" zurück ins DOS XL gehen.
7) "MENU" eingeben, um ins DOS XL-Menü zurückzugelangen.
8) Verwenden Sie das "Initialize Disk"-Kommando mit Option 3, um das DOS.SYS neu auf die Masterdisk zu schreiben.

Anmerkung: Bei Schritt 4 dürfen auch beide Systemvariablen gleichzeitig geändert werden, so daß man nicht alle 8 Schritte zweimal ausführen muß. Beachten Sie bitte auch Anhang A für weitere Informationen hierzu.

7.6 Das Booten direkt in ein BASIC-Programm hinein.

Um unter DOS XL direkt in ein BASIC-Programm hineinzubooten, sind nur einige einfache Schritte durchzuführen, die im einzelnen behandelt werden:

1) Booten Sie Ihre Masterdisk, entweder mit dem Menü oder dem CP.
2) Wünschen Sie das Startup-File auf einer anderen Diskette, so legen Sie diese bitte jetzt ein. Sollte diese Diskette noch nicht initialisiert sein, so beachten Sie bitte Kapitel 3.7 (Menü) oder 6.10 (CP) für Instruktionen zum Formatieren.
3) Nach Erscheinen von 1) (siehe oben) geben Sie bitt:

   ```
   TYPE E: STARTUP.EXC >RETURN<
   ```

   ein, wonach der Bildschirm gelöscht wird und der Cursor in der oberen linken Ecke erscheint.
4) Tippen Sie nun:

   ```
   DO CAR;RUN"D:MENU" >RETURN<,
   ```

   wobei der Filename MENU für den des von Ihnen gewünschten Files stehen soll, das sich natürlich auf der gleichen Disk befinden muß.
5) Geben Sie:

   ```
   >CONTROL< 3
   ```

   ein, d.h. halten Sie CONTROL gedrückt und tippen Sie "3"; dies signalisiert ein "End-of-Line" (Zeilenende). Hierauf wird das File STARTUP.EXC auf die Disk geschrieben und die Kontrolle zurück an das DOS gegeben, was man am Erscheinen des "Dn:" erkennt (Für detailliertere Informationen über .EXC-Files im allgemeinen und STARTUP.EXC-Files im besonderen, sehen Sie bitte in den Kapiteln 8 und 9 nach).

6) Geben Sie DIR ein (oder verwenden Sie die "F"-Option im CP), um ein Inhaltsverzeichnis der Disk zu erhalten. Sollte sich eines der im folgenden aufgeführten Files nicht auf der Diskette befinden, funktioniert das STARTUP-File nicht:

   DOS.SYS
   DO.COM
   STARTUP.EXC
   Ihr BASIC-Programm, dessen Namen Sie vorher angegeben haben.

   Sollte das File DO.COM auf Ihrer Diskette fehlen, so kopieren Sie es bitte mit COPY (oder C) von der Masterdisk. Sollte Ihr BASIC-Programm noch nicht auf der zu erstellenden Disk sein, SAVEn Sie es bitte vom BASIC aus.

7) Bevor Sie nun Ihr Werk durch Aus-und wieder Einschalten des Computers starten, vergewissern Sie sich bitte, daß die BASIC-Cartridge im Slot steckt.

7.7 Das Erstellen einer Masterdisk in doppelter Schreibdichte.

Dieses Kapitel setzt voraus, daß Sie lediglich eine DOS XL- Version in einfacher Schreibdichte besitzen. Einige Diskettenstations-Hersteller liefern zwei Disketten oder eine beidseitige mit den beiden Versionen mit ihren Geräten mit. Sollten Sie schon solch eine Diskette besitzen, brauchen Sie dieses Kapitel nicht zu lesen, sondern booten Sie statt dessen einfach die Version mit der gewünschten Schreibdichte und formatieren eine Disk unter Verwendung des "I"-oder des "D"-Kommandos (Beachten Sie bei Verwendung von "D", daß sie bei der Frage nach der Schreibdichte mit der richtigen Antwort erwidern). Dies gilt natürlich auch für die Kommandos DUPDSK und DUPDBL des Kommando-Prozessors.

Im folgenden wird davon ausgegangen, daß Sie eine Masterdisk in einfacher Schreibdichte gebootet haben; auf Unterschiede zwischen dem Menü-und dem CP-Modus wird entsprechend hingewiesen.

Für die im folgenden beschriebenen Aktionen sollten Sie nicht die Original Masterdisk sondern eine Arbeitskopie verwenden. Da auch ein File umbenannt werden muß, sollte sich kein Schreibschutzaufkleber über der Aussparung befinden.

Vom DOS XL-Menü:

1. Verwenden Sie bitte Option "F" und untersuchen Sie das Inhaltsverzeichnis nach dem File DOSXL.SYS. Erscheint dieses File dort, geben Sie unter Option "U" diesen Filenamen an und verwenden dann Option "R". Hier als "Old name" (Alter Name) DOSXL.SYS und als "New name" einfach DOSXL angeben. Sollten Sie diese Umbenennung durchgeführt haben, so booten Sie jetzt Ihr System neu.
2. Option "X" auswählen und auf den "command:" mit

   INITDBL >RETURN<

   antworten. Auf die Frage "Drive to initialize" (Diskettenstation auf der formatiert werden soll) mit der Zahl 1 gefolgt von >RETURN< antworten. Danach wird man mit "INSERT DISK AND HIT RETURN" (Legen Sie die Diskette ein und drücken Sie RETURN) aufgefordert, die Masterdiskette herauszunehmen und eine leere Disk einzulegen. Drücken Sie nur >RETURN<, wenn Sie dies ausgeführt haben.

INITDBL braucht ca. 1 Minute um die Anforderung zu verarbeiten, daraufhin erscheint das gewohnte "HIT RETURN FOR MENU" (mit RETURN zum Menü), wobei man vor dem Drücken von RETURN zuerst die Masterdisk wieder einlegen sollte.

3) Option "X" anwählen und auf das "command:"

```
SDCOPY *.* -Q
```

eintippen. Das SDCOPY-Programm wird eingeladen und gestartet. Auf die entsprechende Aufforderung hin, legen Sie bitte die zu kopierende Diskette ein und drücken die >RETURN<-Taste.

SDCOPY teilt während des Kopiervorgangs jeweils mit, welches File als nächstes kopiert wird und fragt bei jedem, ob es auch kopiert werden soll. Nach Eingabe von "Y" >RETURN< wird es kopiert, nach "N" >RETURN< nicht. Da davon ausgegangen wird, daß Sie die gesamte DOS-XL-Masterdisk in doppelte Schreibdichte umwandeln wollen, ist das einzige Programm, das nicht kopiert zu werden braucht DOS.SYS, denn dieses wurde bereits von INITDBL auf die Disk geschrieben. Antworten Sie also nur bei DOS.SYS mit "N", ansonsten mit "Y". Wenn Sie natürlich schon vorher wissen, daß Sie für bestimmte Files keine Verwendung haben werden, antworten Sie bei diesen eben entsprechend auch mit "N".

Natürlich ist es eine langwierige Angelegenheit für jedes File die Disketten zu wechseln, aber es lohnt sich.

4) Haben Sie in Schritt 1 die Option "R" benutzt, verwenden Sie diese jetzt auch und geben Sie DOSXL als "Old name" und DOSXL.SYS als "New name" an.

5) Schalten Sie Ihren Computer aus, legen Sie die neue Masterdisk in LW 1 und schalten den Computer wieder an. DOS XL sollte jetzt in doppelter Schreibdichte booten; dies kann man anhand des Inhaltsverzeichnisses überprüfen, denn dort sollte hinter DOS.SYS die Zahl 23 stehen, was bedeutet, daß dieses File jetzt nur noch 23 Sektoren lang ist und die Arbeit erfolgreich war.

6) Haben Sie vorher die Option "R" verwendet, so wählen Sie diese bitte jetzt wieder an und geben Sie für "Old name" DOSXL und für "New name" DOSXL.SYS ~~an, wo~~durch erreicht wird, daß der zusätzlich freigewordene Speicher beim nächsten Booten ebenfalls verfügbar ist.

7) Beschriften und schützen Sie Ihre neue DOS XL-Masterdisk.

Vom DOS XL-Kommando-Prozessor(D1:) aus:
----

1. Verwenden Sie das DIR-Kommando, um festzustellen, ob sich das File DOSXL.SYS auf der Disk befindet, entschützen Sie es mit UNP(rotect) und REN(ame)n Sie es in DOSXL (ohne Extender). Booten Sie jetzt neu.

2. Tippen Sie INITDL ein, um eine leere Diskette in doppelter Schreibdichte zu formatieren und legen Sie die Masterdisk wieder ein, wenn INITDBL fertig ist.

3. Tippen Sie:

   SDCOPY *.* -Q

   ein, um alle Files außer DOS.SYS von der Disk in einfacher auf die in doppelter Dichte umzukopieren. Legen Sie nach getaner Arbeit wieder die Masterdisk in einfacher Dichte ein.

4. Haben Sie DOSXL in Schritt 1 umbenannt, so REN(amen) Sie es jetzt wieder zurück in DOSXL.SYS.

5. Legen Sie die neue Masterdisk in doppelter Dichte ein, booten Sie neu und überprüfen Sie mit DIR, ob nach DOS.SYS die Zahl 23 steht, wenn ja dann waren Sie erfolgreich.

5. Haben Sie DOSXL in Schritt 1 umbenannt, verwenden Sie jetzt wieder REN, um das File auch auf der Disk in DD wieder zurückzubennen.

7. Beschriften Sie Ihre neue Masterdisk und schützen Sie diese mit einem Aufkleber über der Aussparung.

Kapitel 8: Der DOS XL - Boot - Vorgang

Der Einladevorgang von DOS XL in den Computerspeicher ist anders als bei anderen DOS's, etwas komplizierter. Ebenso spielt es eine Rolle welche Files auf einer zu bootenden Disk gelöscht, bzw. hinzugefügt wurden. Um eine Möglichkeit zu liefern diesen Bootprozess für eigene Anwendungen entsprechend umzugestalten, wird im folgenden der ganze Vorgang genau beschrieben.

8.1 DOS - Systeme mit erweitertem Speicherbereich und DOSXL.SYS

Auf der Masterdisk befinden sich zwei spezielle Files: DOSXL.XL und DOSXL.SYS, die zusammen als das erweiterte Speicher-DOS-System bezeichnet werden.

Um die Vorteile eines solchen Systems voll nutzen zu können, braucht man folgendes:

1. Einen ATARI-Computer der XL-Serie mit 64Kbyte RAM (1200XL, 800XL, oder einen erweiterten 600XL etc.)
2. Eine OSS-Super Cartridge (ACTION!, BASIC XL, MAC/65 etc.)

Haben Sie keinen der oben genannten Punkte, so können Sie gleich zu Kapitel 8.2 weiterblättern.

In der Grundeinstellung der Masterdisk sind diese extra Speicherbereiche nicht zugänglich. Möchten Sie diese Vorteile nutzen, so lesen Sie bitte hier weiter, ansonsten in Kapitel 8.2.

Wenn Sie die OSS SuperCartridge oft verwenden, sollten Sie das File DOSXL.SUP wie unten beschrieben umbenennen. Benutzen Sie dieses Modul nicht, haben aber einen XL-Computer, sollten Sie DOSXL.XL wie folgt umbenennen:

Arbeiten Sie mit dem DOS XL-Menü, so wählen Sie bitte Option "U" aus, und geben zum Entschützen einen der beiden vorher gennanten Filenamen an. Dann "R" anwählen, und für "Old name" den entsprechenden Namen angeben, den Sie auch vorher unter "U" eingetippt haben. Auf "New name" antworten Sie bitte mit DOSXL.SYS und gehen zurück ins Menü.

Oder: Im CP mit UNP das gewünschte File entschützen und mit REN entsprechend umbenennen.

Wenn Sie nun Ihren Comuter aus-und wieder einschalten, wird das erweiterte Speicher-DOS-System gebootet.

Wie funktioniert das ?

Während das DOS der meisten anderen Hersteller nur im File DOS.SYS liegt, belegt es bei DOS XL zwei separate Files. Das erste, DOS.SYS muß sich auf jeder Disk befinden, um diese bootbar zu machen; es wird zu Beginn des Bootprozesses in den Speicherberiech von $700 bis $1E00 geladen.

Nun überprüft dieses DOS (es ist in sich wirklich ein eigenständiges DOS), ob sich das File DOSXL.SYS auf der gebooteten Disk befindet. Ist das der Fall, dann wird angenommen, daß DOSXL.SYS ein erweitertes DOS-System enthält und dieses eingeladen. Nachdem das DOS XL die Kontrolle übernommen hat, geschehen mehrere Dinge.

Zuerst wird überprüft, ob Sie wirklich die Hardwarevoraussetzungen besitzen, die Sie durch das vorhergegangene Umbenenen angegeben haben. Sind die Hardwarevoraussetzungen nicht gegeben, so wird die Kontrolle wieder an das normale DOS zurückgegeben und nichts weiter passiert; d.h. das DOS XL steht jetzt in keinem Fall zur Verfügung.

Ist jedoch alles in Ordnung, so schiebt sich das DOS XL in den RAM-Speicher, entweder "unter" die OSS SuperCartridge, oder unter das ATARI-Betriebssystem.

Dieser neu geladene Code ist nun das DOS des Computers, wodurch 3 bis 5K Byte zusätzlich gespart werden und wobei ein Speicherbereich belegt wird, der von der Super C. durch Bankswitching, (d.h. vertauschen von Speicherbereichen) oder vom ATARI-OS durch speziell bereits in die XL-Modelle eingebaute Hardware ansprechbar wird.

Beachten Sie bitte, daß Sie das File DOSXL.SYS wieder umbenennen können (am besten in DOSXL ohne Extender), wenn Sie diesen extra Speicherbereich nicht verwenden wollen.

8.2 Das AUTORUN.SYS - File

Während des Bootvorganges und wenn das DOS XL entweder geladen, unterdrückt oder nicht verwendet wurde (siehe Kapitel 8.1), wird die Disk nach einem File mit Namen AUTORUN.SYS durchsucht (ein solches File befindet sich nicht auf der Masterdisk). Wird es gefunden, so lädt das DOS es genau wie bei der Ausführung eines "Load binary"-Kommandos ein.

Möchten Sie z.B. das File RS232.COM beim Booten gleich ausführen, so ist dieses lediglich in AUTORUN.SYS umzubenennen (Beachten Sie bitte auch das Kapitel über RS232FIX.COM).

Der Einladevorgang eines AUTORUN.SYS-Files unter DOS XL ist identisch mit dem unter den ATARI-DOS's.

8.3 Das STARTUP.EXC - File

Es gibt noch einen anderen Schritt während des Bootvorganges:

Wenn kein AUTORUN.SYS - File auf der Disk gefunden wird, oder wenn die Kontrolle durch einen 6502 RTS-Befehl dem DOS zurückgegeben wird, dann sucht das DOS XL ein STARTUP-File auf der gebooteten Diskette. Ein solches File ist ein Textfile und enthält Befehle, die dem DOS XL Kommando-Prozessor übergeben werden.

Auf der DOS XL - Masterdisk befindet sich ein STARTUP.EXC-File, das nur REM-Kommandos, die Bildschirmmitteilungen und das Kommando MENU enthält, das das Menü startet (im nächsten Kapitel wird eine andere Möglichkeit gezeigt das Menü zu laden).

Bemerkung am Rande: Um den Inhalt eines solchen Files zu ändern, Option "Copy Files" auswählen und "E:" und "D:STARTUP.EXC" als "From" und "To" angeben. Nachdem der Bildschirm gelöscht wurde und der Cursor wieder am linken oberen Rand erschienen ist, können die gewünschten Befehle (einer in eine Zeile) eingegeben werden. Sind alle Kommandos eingetippt, die Eingabe mit >CONTROL< 3 abschließen (CONTROL gedrückt halten und 3 tippen). Die Anweisungen werden dann in das File STARTUP.EXC geschrieben.

Möchten Sie kein STARTUP.EXC verwenden, löschen oder renamen Sie es einfach (z.B.: STARTUP.TXT als Textfile).

Einige auf Modulen basierende Produkte (wie z.B. ATARI-Schreiber) arbeiten nicht einwandfrei, wenn sich ein STARTUP.EXC-File auf der zu bootenden Disk befindet. Erstellen Sie sich für solche Fälle eine extra Bootdisk wie folgt:

1) Duplizieren Sie die Masterdisk auf eine neue, leere Diskette.
2) Löschen Sie hier das File STARTUP.EXC.
3) Löschen Sie das File MENU.COM (nur, wenn Sie mehr Speicher zur Verfügung haben möchten).

Nun kann diese Diskette zum Booten von z.B. ATARI-Schreiber verwendet werden.

8.4 Das MENU.COM - File

Als letzten Schritt des DOS XL -Bootvorganges und vorausgesetzt, daß sich weder ein AUTORUN.SYS- noch ein STARTUP.EXC-File auf der Disk befinden, wird nach dem File MENU.COM gesucht, das, wenn es gefunden wurde, an der niedrigsten freien Adresse im Speicher eingeladen (den aktuellen MEMLO-Zeiger in den Speicherzellen $2E7 und $2E8) und dem am Ende des Bootprozesses die Kontrolle übergeben wird.

Ist ein Modul eingesteckt, wird dieses jetzt angesprungen, andernfalls wird das DOS XL-Menü oder der CP aktiviert.

Kapitel 9: Batchprocessing

9.1 überblick

Man wird oft feststellen, daß man eine bestimmte Kommandofolge häufiger verwendet. Das DOS XL ermöglicht es solche Kommandos in einem File mit besonderen Eigenschaften abzulegen, das man mit einem einzigen Kommando aufrufen kann. Worauf alle in diesem gespeicherten File befindlichen Kommandos ausgeführt werden. Dadurch läßt sich natürlich ein Haufen Zeit und Tipparbeit sparen.

Nehmen wir an, Sie haben einige ACTION!-Programme geschrieben, die hintereinander ablaufen sollen. Dies kann man auf zwei verschiedene Weisen lösen:

1) Für jedes Programm den externen CP-Befehl einen nach dem anderen eingeben. Brauchen die Programme in ihrer Ausführung länger, so kann es vorkommen, daß man Stunden warten muß, nur um einen neuen Programmnamen anzugeben.

oder:

2) man erstellt ein Batch-File, das die zum Nacheinanderablaufen der Programme erforderlichen DOS XL-Kommandos beinhaltet und gibt dann nur ein Kommando ein, wodurch man Zeit gewinnt für andere wichtigere (oder lustigere) Dinge.

Die zweite Methode ist offensichtlich praktischer und läßt sich auch mehrmals hintereinander verwenden.

Jedes Textfile mit dem Extender .EXC kann als DOS XL-Batch-Ausführungs-File verwendet werden. Die Ausführung eines solchen Files wird wie die eines externen Befehls eingeleitet, außer, daß ein Klammeraffe ("@") vorangestellt werden muß. Um z.B. das EXECUTE-File DEMO.EXC in D1: zu starten, ist folgendes einzutippen:

```
D1:@DEMO
```

Der CP öffnet ~~darauf~~hin einen Kanal für das File D1:DEMO.EXC für Eingabe und weist das DOS XL an, dieses File Zeile für Zeile zu lesen und die CP-Kommandos genauso als wären sie mit der Tastatur eingegeben worden, auszuführen.

9.2 Das Format eines EXC-Files.

Ein Execute-File ist einfach ein Textfile, bei dem jede Zeile bei der Ausführung wie ein CP-Kommando behandelt wird.

Die drei Grundregeln für Textfiles sind:

1) sie müssen erlaubte DOS XL Konsolenprozessor-Kommandos enthalten,
2) sie müssen kürzer als 128 Zeichen sein,
3) sie müssen mit einem carriage-return (Zeilenvorschub/ATASCII: $9B) beendet sein.

DOS XL erlaubt es, daß vor den Kommandos Zahlen oder Leerstellen verwendet werden, so daß die einzelnen Kommandos zur Erhöhung der Lesbarkeit nummeriert und kommentiert werden können.

Die Kommando-Files-Zeilen:

```
          LOAD OBJ.TEST       >RETURN<
und       100 LOAD   OBJ,TEST >RETURN<
```

werden von DOS XL gleich behandelt. Der CP durchsucht eine Zeile nach dem ersten Zeichen, das keine Zahl und keine Leerstelle ist, bevor das eigentliche Kommandowort untersucht wird. Es kann also jeder Editor, einschließlich der des MAC/65 zur Erstellung von EXEC-Files verwendet werden.

Anmerkung: Man kann ein EXECUTE-File (oder jedes andere Textfile) auch mit "TYP E: <diskfile> erstellen. (TYPE löscht den Bildschirm, worauf man seinen Text Zeile für Zeile eingeben kann. Der Kopiervorgang wird mit Drücken von >CONTROL< 3 eingeleitet).

9.3 Interne Kommandos für EXC-Files

DOS XL stellt vier spezielle interne Kommandos extra zur Verwendung mit EXC-Files zur Verfügung:

REMARK	Bemerkung oder Kommentar (macht sonst nichts)
SCREEN	schaltet das EXECUTE-Echo ein, d.h. die gerade ausgeführten Befehle werden am Bildschirm ausgegeben (Grundeinstellung).
NOSCREEN	Schaltet das Echo aus.
END	~~Stoppt~~ die Ausführung eines EXECUTE-Files und geht zurück zum CP.

Weiter Informationen in Kapitel 5

9.4 Das Stoppen eines Batch-Files

Während der Ausführung eines BATCH-Files können ja einmal Umstände auftreten, die es erforderlich machen dieses zu stoppen, was bei Systemfehlern oder Fehlern in einem Userprogramm passieren kann.

9.4.1 Stops durch DOS XL selbst.

In den Augen eines Computers sind Menschen nicht ganz perfekt und machen oft Fehler. Falsche DOS XL-Kommandos verursachen Fehlermeldungen. Stellt das DOS XL bei der Ausführung eines Batch-Files einen Fehler fest, wird die Fehlermeldung wie üblich angezeigt und das File gestoppt. Dieser Stop wird aber nur durchgeführt, wenn DOS XL den Fehler entdeckt, nicht wenn der Fehler im Userprogramm liegt.

Die Ausführung eines .EXC-Files wird auch nach einem CARtridge-Kommando beendet.
Schließlich stoppt das File auch, wenn sein Ende erreicht ist.

9.4.2 Stops durch das Userprogramm.

Manchmal ist es notwendig, daß ein Userprogramm die Ausführung anhält. Die häufigste Ursache ist die, daß das Programm einen aufgetretenen Fehler, der den Ablauf der folgenden Programme beeinträchtigen würde, entdeckt hat. Die weitere Ausführung eines Batch-Files hängt von einem einzigen DOS XL-Flag (Markierung) ab. Setzt ein Userprogramm dieses Byte auf 0, so wird die Ausführung nach der Rückkehr ins DOS sofort gestoppt. Das EXECUTE-Flag liegt genau 12 Bytes hinter dem Anfang von DOS XL, das ab Speicherstelle 10 ($0A) beginnt. Der folgende kleine BASIC-Programm-Ausschnitt würde das EXECUTE-File ausschalten und die Kontrolle zurück ans DOS XL geben:

```
1000 CPADR=PEEK(11)=*256+PEEK(10)
1010 EXCFLG = CPADR+1
1020 POKE EXFLG,0
1030 DOS
```

oder vom BASIC XL aus:

```
100 Poke Dpeek(10)+11,0 :Dos
```

Beachten Sie bitte, daß ein CAR-Kommando automatisch die Ausführung eines Batch-Files stoppt, so daß ein BASIC-Programm nur zur Veranschaulichung dienen kann.

9.5 STARTUP.EXC - ein spezielles File.

Der Execute-Filename STARTUP.EXC hat unter DOS XL eine spezielle Bedeutung. Wenn das System beim Einschalten gebootet wird, durchsucht das DOS XL das Inhaltsverzeichnis einer Diskette nach dem File STARTUP.EXC und führt dieses aus, bevor es irgendwelche Tastaturkommandos entgegennimmt.

Siehe auch Kapitel 8.3

9.6 Wie ein EXECUTE-File funktioniert.

Wenn man das Kommando "@Filename" eingibt, speichert der CP den Filenamen in einem internen Buffer (CPEXFN) und setzt ein Flag (CPEXFL), um anzuzeigen, daß eine Batchverarbeitung läuft.

Jedesmal bevor der Benutzer mit der Meldung D1: nach einer Eingabe gebeten wird, überprüft der CP dieses Flag. Wenn es gesetzt ist, wird ein Kanal für das Batch-File geöffnet (mit dem vorher angegebenen Filenamen) und, wenn das File nicht zum ersten Mal zur Batchverarbeitung geöffnet wird, ein Zeiger (wieder mit dem Flag CPEXFL) auf den Anfang der nächsten Befehlszeile festgelegt.

Diese Zeile wird dann als nächstes in den Kommando-Buffer eingelesen. Nun wird wieder die neue Position in einer internen Variablen von CPEXNP festgehalten und auf den folgenden Zeilenanfang gedeutet.

Schließlich wird die im Kommandobuffer befindliche Anweisung ausgeführt, genauso als wenn diese von Hand eingegeben worden wäre.

Ist der Befehl ordentlich ausgeführt worden und mit einem RTS- oder einem JMP-Befehl durch DOSVEC abgeschlossen, wiederholt sich der gesamte Vorgang bis das EXECUTE-Flag irgendwie ausgeschaltet wurde.

Dem erfahrenen Programmierer wird schon aufgefallen sein, daß man durch ändern der Variablen von CPEXxx interessante Möglichkeiten zur Verfügung hat. Diese Speicherstellen sind alle im File SYSEQU.ASM definiert und sind Offsets von der Adresse in DOSVEC (Speicherstelle $000A). Ein Beispiel hierfür ist der programmkontrollierte Stop aus Kapitel 9.4.2.

Beachten Sie bitte auch Kapitel 10.2.3, in dem auch einiges hierzu behandelt wird.

Kapitel 10: Maschinensprache und DOS XL.

Wie bereits in Kapitel 1.6 angedeutet, handelt es sich bei DOS XL um ein strukturiertes und nach außen hin offenes System. Applikationsprogramme (oder auch Sprachen wie z.B. BASIC XL) können ohne Schwierigkeiten auf das Betriebssystem durch den System-Aufruf-Vektor (mit "CIO" im File SYSEQU.ASM bezeichnet) zugreifen. Der CIO entscheidet dann welcher Geräteverwalter den I/O-Request (die Ein-/Ausgabe-Anforderung) erhält und verwaltet einen Hautpteil des laufenden Programmes.

Legt sich ein Programm auf bestimmte Aufrufe des CIO fest und verwendet die in SYSEQU.ASM angegebenen Labels (Bezeichnungen), so kann es problemlos zwischen den verschiedenen DOS XL-Versionen ausgetauscht werden.

Auf jeden Fall folgt jetzt eine Beschreibung der angewandten Maschinensprache-Aufruf-Sequenzen und Parametern unter DOS XL.

10.1 Schnittstelle zur I/O-Routine.

10.1.1 Die Struktur des IOCB's.

Ruft ein Programm durch das Betriebssystem den CIO, erwartet das OS die Adresse eines ordentlich formatierten IOCB's (Input Output Control Block = Ein-/Ausgabe Kontroll-Block). Der Einfachheit halber wurden 8 vordefinierte IOCB's, jeder 16 Bytes lang, bereitgestellt und das rufende Programm spezifiziert den gewünschten durch übergabe einer IOCB-Nummer*16 in das X-Register des 6502. So müßte z.B., um auf die IOCB-Nummer 4 zuzugreifen das X-Register zur übergabe an das OS die Zahl $40 enthalten. Beachten Sie bitte, daß die IOCB-Nummer im direkten Verhältnis mit der Kanalnummer im BASIC (z.B. wie bei PRINT #6, etc.) steht. Die IOCB's liegen im Speicherbereich von $0340 bis $03BF im ATARI (Sie sollten sich aber lieber auf die Angaben in SYSEQU.ASM als auf Speicheradressen verlassen).

Wird die Kontrolle an das OS gegeben, so verwendet dieses den Wert im X-Register, um den gewünschten IOCB zu finden und festzustellen, was der Benutzer will. In Tabelle 10.1 stehen die DOS XL-Standardnamen für jedes Feld im IOCB mit einer kurzen Beschreibung des Zweckes (Bitte schauen Sie sich die Tabelle an, bevor Sie weiterlesen).

Das Userprogramm sollte die Felder ICHID, ICDNO, ICSTA und ICPUT niemals verändern, da diese vom OS gesetzt werden. Außerdem sollten ICAUX1 bis ICAUX6 nicht modifiziert werden, wenn das spezielle Device oder der I/O-Request es nicht unbedingt erfordern. Das wichtigste Feld ist das ein-byte-Feld ICCOM, das dem OS mitteilt, welche Funktion gewünscht wird.

Tabelle 10.1

IOCB STRUKTUR

Feld Name	Offset im IOCB (bytes)	Größe d.Feldes (bytes)	Zweck des Feldes
ICHID	0	1	wird vom OS gesetzt, Index in die Device-Namenstabelle für momentan geöffnete Files, Inhalt $FF, wenn kein F. für diesen IOCB geöffnet ist.
ICDNO	1	1	vom OS gesetzt; Device Nummer (d.h. 1 für D1:xxx" oder 2 für D2:yyy")
ICCOM	2	1	Kommando-Anforderung des User-Programmes. Definiert, wie der Rest des IOCB's formatiert wird
ICSTA	3	1	vom OS ges.;Letzter vom Device geantworteter Status, aber nicht unbedingt der Status, der über die STATUS-Kommando-Anforderung ermittelt wird
ICBADR	4	2	Bufferadresse.Zwei-Byte Adresse in normaler 6502 high/low Ordnung; gibt die Adresse des Buffers für den Datentransfer oder die Adresse des Filenamen für OPEN, STATUS etc.
ICPUT	6	2	vom OS gesetzt; Adr. minus 1 der Device one-byte Routine. Nützlich, wenn Hochgeschwindigkeits Ein-byte Datentransfers benötigt werden.
ICBLEN	8	2	Buffer-Länge. Gibt die max. Anzahl von zu transferierenden Bytes bei PUT/GET Operationen an; wird um 1 vermindert für jeden Byte-Transfer

ICAUX1	10	1	Hilfs-byte Nummer 1; wird bei OPEN zur Spezifizierung des Filezugriffs benötigt. Einige Drivers machen zusätzlichen Gebrauch von diesem Byte
ICAUX2	11	1	Hilfs-byte Nummer 2; einige serielle Port-Funktionen können Gebrauch von diesem Byte machen; dieses und die folgenden Bytes sind für spezielle Anwendungen von den Device Drivern verwendet.
ICAUX3 ICAUX4	12	2	Nur für Diskfiles, bei denen die Sektorennummer von NOTE und POINT übergeben wird. (Diese Bytes können auch separat von anderen Diskdrivern verwendet werden.)
ICAUX5	14	1	Nur für Diskfiles; die Byte-im-Sektor Nummer, die von NOTE und POINT übergeben wird.
ICAUX6	15	1	Ein selten benutztes Hilfsbyte

Tabelle 10.2 IOCB Feld benutzen

IOCB Feld Name	0	1	2	3	4 / 5	6 / 7
KOMMANDOTYP	ICHID	ICDNO	ICCOM	ICSTA	BUFFER ADRESSE ICBADR	LEGE IN ADRESSE BYTE ICPUT
OPeN	*	*	3	*	Filename	*
CLOSE	*		12	*		
dynamischer STATUS		*	13	*	Filename	
Get TeXT Record			5	*	Buffer	
Put TeXT Record			9	*	Buffer	
Get BINary Rec.			7	*	Buffer	
Put BINary Rec.			11	*	Buffer	
Externe Kommandos: Nur für Diskfile-Manager						
REName		*	32	*	Filename	
ERAse		*	33	*	Filename	
PROtect		*	35	*	Filename	
UNProtect		*	36	*	Filename	
NOTE			38	*		
POINT			37	*		

Legende: '*' —— wird vom OS gesetzt, wenn dieses Kommando gewählt wurde.
'Buffer' Adresse des Datenbuffers.
'Filename' Adresse des Filenamen.

8 ! 9	10	11	12	13	14	15	IOCB Feldname	
Buffer Länge ICBLEN	ICAUX 1	ICAUX 2	ICAUX 3	ICAUX 4	ICAUX 5	ICAUX 6	wie in SYSEQU angegeben Kommandonamen	
	mod						COPN	
							CCLOSE	
							CSTAT	
Länge							CGTXTR	
Länge							CPXTR	
Länge							CGBINR	
Länge							CPBINR	
(siehe Kapitel 10.1.2)								
							CREN	
							CERA	
							CPRO	
							CUNP	
			sec num		byt		CNOTE	
			sec num		byt		CPOINT	

<u>Legende:</u> 'Länge' Länge eines Datenbuffers
'mod' OPEN-Modus (lesen, schreiben etc.)
'sec num' Sektornummer (siehe Kap.10.1.2)
'byt' Byte im Sektor(siehe Kap.10.1.2)

10.1.2 Die Ein-/Ausgabe-Kommandos

In Tabelle 10.2 ist eine Zusammenstellung von IN/OUT-Kommandos und ihre Verwendung in den verschiedenen IOCB-Felder aufgelistet. Die ersten sieben orientieren sich an DOS XL und werden in Teil A) dieses Kapitels behandelt; die anderen sechs sind Filemanager und folgen in Teil B).

Die meisten Kommandos manipulieren ein Device in irgendeiner Art, deshalb wird kurz auf diese eingegangen. Device-Namen unter DOS XL sind ziemlich einfach gehalten; sie bestehen aus einem einzelnen Buchstaben, optional gefolgt von einer nummerischen Angabe, mit der zwischen verschiedenen gleichen Peripheriegeräten unterschieden wird, und einem Doppelpunkt. Die folgenden Device-Namen werden unter DOS XL und ATARI-DOS verwendet:

E: Tastatur/Bildschirm-Device. Normaler Konsolen-Output.

K: Tastatur-Device

S: Bildschirmdevice. Buchstaben oder Graphikzeichen.

P: Printer (Drucker). Der Standard-Device-Driver erlaubt nur einen Drucker.

C: Cassettenrecorder.

D: Diskfile-Manager; hier wird normalerweise noch ein Filename benötigt.

Andere Device-Namen sind möglich (z.B. für ein RS232-Interface) und die Tatsache, daß sich beliebige weitere Devices implementieren lassen, ist ein Zeichen dafür, daß DOS XL ein richtiges Betriebssystem ist. Auf die Struktur der Device-Driver wird in einem späteren Kapitel (10.3) noch ausführlicher eingegangen, jedoch sollte vorweg schon einmal bemerkt werden, daß das OS ROM alle oben genannten Driver außer D: beinhaltet, wodurch 5K Speicher belegt werden, von denen alleine der Screenhandler mit seinen Editier-und Graphik-Möglichkeiten 3K belegt.

A) Die Standard-DOS XL-Kommandos.

Das Betriebssystem selbst versteht nur einige fundamentale Kommandos, DOS XL benötigt aber zusätzlich noch externe Kommandos für einige Devices (XIO in BASIC). Im folgenden werden diese fundamentalen Kommandos kurz beschrieben:

OPEN

öffne einen Kanal (oder Device, File IOCB) zum Schreiben und/oder Lesen. Das OS erwartet in ICAUX1 eine Byte, das den Modus bestimmt:

ICAUX1	Modus
4	Nur Lesen
6	Nur Directory lesen
8	Nur Schreiben
9	Nur Schreiben und Anhängen
12	Schreiben/Lesen (Update)

Auch der Name des Devices (und bei Disk auch der Filename) müssen dem OS übergeben werden, was durch Plazieren der Adresse eines Strings in ICBADR geschieht.

CLOSE

Beenden eines Device-Zugriffs; ohne weitere Angaben

STATUS

Fordert den Status eines Device/Files an. Das Device kann diese Anforderung nach eigenem Wunsch verarbeiten und (hoffentlich) eine ordentliche Antwort zurückliefern. Wie bei OPEN, muß auch hier die Adresse des Filenamen in ICBADR abgelegt sein.

GET TEXT

ist ein vielseitiges Kommando; veranlaßt das OS ein Byte nach dem anderen von einem Device/File, das bereits geöffnet sein muß, zu holen, bis entweder der zur Verfügung gestellte Bufferspeicher erschöpft, oder ein RETURN-Zeichen (ATARI $9B) aufgetaucht ist. Der Benutzer kann den Bufferspeicher durch Ablegen seiner Adresse in ICBADR und seine maximale Länge in ICBLEN, festlegen.

PUT TEXT
----
analog zu GET TEXT gibt das OS ein Byte nach dem anderen aus, bis ein RETURN aufgetaucht, oder der Speicher leer ist. Benötigt ICBADR und ICBLEN zur Spezifizierung.

GET DATA

Extrem flexibles Kommando; veranlaßt das OS von einem vorher geöffneten Device/File, die Anzahl der, von ICBLEN spezifizierten Bytes in den, von ICBADR festgelegten Buffer, einzuladen. Es werden keinerlei Kontrollen über den Inhalt der transferierten Daten durchgeführt.

PUT DATA

Revers zu GET DATA; d.h. das OS gibt die ICBLEN-Daten, aus dem ICBADR-Buffer aus. Auch hier keine Inhaltskontrollen.

B) Kommandos in Verbindung mit dem Filemanager-System.

In Tabelle 10.2 stehen noch einige DOS XL-Kommandos, auf die noch nicht eingegangen wurde. Auf diese externen Kommandos wird via die externe Anforderungs-Routine in der Device-Driver Benutzungstabelle zugegriffen (Siehe Kapitel 10.3 für Details über Device-driver). Diese Anweisungen, die für das Disk-Device im Filemanager-System enthalten sind, sind es auf jeden Fall wert, ihnen ein extra Kapitel zu widmen:

ERASE, PROTECT und UNOROTECT

Auch bekannt als Delete, Lock und Unlock, versorgen diese Kommandos das OS einfach mit einer Kanalnummer (d.h. das X-Register enthält die IOCB-Nummer*16), eine Kommandonummer (ICCOM) und einen Filenamen (in ICBADR). Wenn das OS also die Kontrolle an das FMS übergibt, so wird der Versuch unternommen, die entsprechende Anforderung zu verarbeiten. Beachten Sie bitte, daß der Filename wildcards enthalten darf.

RENAME

Sehr ähnlich zu ERASE auch in der Verwendung. Der einzige Unterschied liegt in der Form des Filenamen. Richtige Form: (Dn:)altname.ext,neuname.ext. Die Disk-Spezifizierung wird und darf nur einmal angegeben werden.

NOTE und POINT

Anders als bei OPEN sind dies die einzigen beiden Kommandos im Standard-DOS XL, die die AUXiliary (Hilfs-) Bytes des IOCB benutzen. Für diese Anweisungen gibt der Benutzer die Kanal- und die Kommandonummer an und erhält dann File-Zeiger-Informationen über die drei AUX-Bytes. ICAUX3/ICAUX4 werden als konventionelle 6502 LSB/MSB 16-bit Integer verwendet: diese spezifizieren den aktuellen (NOTE) oder den aktuell-zu-machenden (POINT) Sektor innerhalb eines bereits geöffneten Diskfiles. ICAUX5 ist ähnlich dem aktuellen (NOTE) oder dem aktuell-zu-machenden (POINT) Byte innehalb eines Sektors.

FMS Erweiterungen zum OPEN-Befehl.

OPEN ist nicht wirklich eine externe Operation, aber für den Disk-I/O muß man wissen, daß das FMS zwei weitere Modi neben den fundamentalen OS-Modi beherrscht.

Enthält ICAUX1 eine 6 bei einer OPEN-Anforderung unter DOS XL, wird das Disketteninhaltsverzeichnis zum Nur-Lesen geöffnet (statt eines Files). Die Adresse ICBADR spezifiziert nun das File (oder die Files, wenn wildcards verwendet wurden), das als Teil eines Gesamtinhaltsverzeichnisses gelistet werden soll. Beachten Sie bitte, daß das FMS nach einem solchen OPEN eine Verwendung des GETREC-OS-Aufrufs erwartet.

Enthält ICAUX1 eine 9, wird das spezifizierte File als Nur-Schreib-File geöffnet, aber der File-Zeiger wird auf das Ende des Files gesetzt.

10.1.3 Ausgegebene Fehlermeldungen.

Beim Rücksprung aus jedem OS-Aufruf enthält das Y-Register den Beendigungs-Code der aufgerufenen Operation. Code 1 zeigt einen normalen Status an, d.h. daß alles erfolgreich verlaufen ist. Die Codes von $02 bis $7F (2 bis 127 hexadezimal) bedeuten "Warnung"; die von $80 bis $FF (128 bis 255 dezimal) sind "hard"-errors. Folgende Assembler-Sprache-Sequenz entspricht dem:

```
JSR CIOV ; OS aufrufen
TYA      ; Fehlercode überprüfen
BMI OOPS ; wenn $80-$FF, dann Fehler.
```

Theoretisch sind diese Status-Codes wie TYA, mit denen sich das DOS XL zurückmeldet, unnötig. In der Praxis stimmt das vielleicht auch, denn eine kleine Paranoia verlängert oft das Leben sowohl menschliches als auch programmiertes.

10.2 Manipulierung von DOS XL.

Diejenigen, die in Maschinensprache programmieren werden eigentlich fast immer mit dem OS in irgendeiner Weise Kontakt aufnehmen müssen. Soll der Assembler-Code ein externes Kommando werden, so wird es nötig sein eine Schnittstelle zum DOS XL zu haben (Weiter Informationen zum OS-Interface in Kap. 10.1).

Sollten Sie Sofware schreiben wollen, die dazu bestimmt ist, mit DOS XL zusammenzuarbeiten, werden Sie bestimmt verschiedene Speicherstellen untersuchen und/oder modifizieren müssen, oder Zugriff auf verschiedene Routinen innerhalb von DOS XL benötigen.

10.2.1 SYSEQU.ASM

Jede DOS XL-Masterdisk enthält ein Assembler-Quellfile mit Namen SYSEQU.ASM, das verschiedene gemeinsame Systemteile vom ATARI-OS und DOS XL beinhaltet; dieses File kann im MAC/65 mit der INCLUDE-Funktion in ein Maschinensprache-Programm eingebunden werden (.INCLUDE #D1:SYSEQU.ASM). Auf jeden Fall muß es in den MAC/65 geENTERT und dann zurück auf die Disk geSAVEt werden, da es sich in Textform befindet.

10.2.2 Speicherstellen des Kommando-Prozessors.

Der Kommando-Prozessor (CP) wird im Speicher gleich hinter dem Filemangager-System (FMS) plaziert, wenn die DOS.SYS-Version von DOS XL gebootet wurde. Da die genaue Speicherstelle bei verschiedenen Versionen des FMS und/oder durch verschiedene Speicherkonfigurationen etwas variieren kann, wurde eine bestimmte Stelle festgelegt, die auf den CP zeigt: Die Speicherstelle CPALOC ($0a) enthält die Adresse des DOS XL und CP-Warmstart-Eintritts-Punktes. Die meisten ATARI-Programme sollten in den CP durch JMP zu der in CPALOC angegebenen Adresse, zurückkehren.

10.2.3 EXECUTE-Parameter.

Das CP EXECUTE-Flag liegt in CPEXFL ($08) vom Anfang des CP. CPALOC kann als indirekter Zeiger zum Zugriff auf das EXEC-Flag verwendet werden:

```
LDY   #CPEXFL    ;Hole DISPL ins FLAG
LDA   (CPALOC),Y ; Lade Flag
```

Das EXEC-Flag hat vier bits, die den Ausführungs-Vorgang kontrollieren:

Name	Bit #	
EXCYES	$80	Wenn 1, dann läuft ein EXECUTE.
EXCSCR	$40	Wenn 1, dann wird das Kommando-Echo am Bildschirm unterdrückt.
EXCSUP	$20	Wenn 1, dann beginnt ein Kaltstart-Execute; wird verwendet, um zu vermeiden, daß ein "File not found" Fehler auftritt, wenn sich das File STARTUP.EXC nicht auf der Disk befindet.
EXCNEW	$10	Wenn 1, startet ein neuer EXECUTE; teilt dem CP mit, mit der ersten Zeile des Files zu beginnen.

Der CP beginnt einen EXECUTE mit dem öffnen eines Files, POINTed auf die nächste Zeile, READs eine Zeile, NOTEd die neue nächste Zeile und CLOSEd das File. Um diese Funktionen auszuführen, muß der CP den EXECUTE-Filenamen und die drei Byte NOTE-Werte speichern. Der Filename wird in CPEXFN ($0C) und die drei NOTE-Werte in CPEXNP ($1C) im CP gespeichert (CPEXNP = ICAUX5; CPEXNP+1 = ICAUX4; CPEXNP+2 0 ICAUX3). Indem man die verschiedenen Execute-Parameter ändert, kann man das Aneinanderketten von Executefiles, das überspringen verschiedener Zeilen in einem File etc., beeinflußen.

10.2.4 Voreingestellte Drive-Speicherstellen.

Die im CP voreingestellte Drive-Filespezifizierung liegt in CPDFDV ($07 im DS/A+). Das Drive in der Grundeinstellung ist ATASCII Dn:, wobei 'n' die ATASCII-Grund-Drivenummer ist.

10.2.5 Externe Parameter.

Die externen Kommandos können durch Parameter, die in die Kommandozeile geschrieben werden, aufgerufen werden. Das CP-Kommando:

D1:COPY QUELLFILE D2:ZIELFILE

wäre ein Beispiel hierfür. Die gesamte Kommandozeile wird im Eingabebuffer in CPCMDB ($3F Bytes im CP) gespeichert und ist für den Benutzer verfügbar. Da die meisten Kommandoparameter Filenamen sind, beinhaltet der CP eine Funktion, die diese Parameter als Filenamen behandelt; diese Routine beginnt bei CPGNFN ($03 Bytes im CP), holt den nächsten Parameter und verschiebt diesen in den Filenamen-Buffer in CPFNAM ($21 Bytes im CP). Enthält der Parameter keine Device-Spezifikation, wird der voreingestellte Drive vom CP vorangestellt. Das erste Mal, wenn COPY CPGNFN aufruft, wird die Filekennzeichnung "D1:QUELLFILE" in CPFNAM abgelegt. Wenn COPY CPGNFN zum zweiten Mal aufruft, wird die Filebezeichnung "D2:ZIELFILE" an CPFNAM übergeben. Wird CPGNFN noch öfters aufgerufen, so erscheint in CPFNAM immer die Filespezifikation der Grundeinstellung. Um das Ende der Parameterkette herauszufinden, kann man CPBUFP ($0A im CP) überprüfen. Wenn CPBUFP den CPGNFN-Aufruf nicht verändert, dann sind keine Parameter mehr vorhanden. Der Filename-Buffer wird immer auf 16 Bytes mit ATASCII EOL ($9B)-Zeichen aufgefüllt. Das folgende Beispiel erzeugt einen Vektor, der die GET Filename-Routine aufruft:

```
CLC
LDA    CPALOC     ;CPGNFN  hinzufügen
ADC    #CPGNFN    ;CPALOC Wert dazuzählen
STA    GETFN+1    ;und im
LDA    CPALOC+1   ;Adressenfeld
ADC    #0         ;der Sprung-Anweisung
STA    GETFN+2    ;platzieren.
GETFN  JMP    0
```

Die folgende Routine holt den nächsten Filenamen nach CPFNAM:

```
LDY    #CPBUFP    ;speicher CPBUFP
LDA    (CPALCO),Y ;Wert
PHA
JSR    GETFN      ;nächsten File-Parameter holen
LDY    #CPBUFP
PLA               ;Test, ob keiner mehr da ist.
CMP    (CPALOC),Y ;Parameter
BEQ    NONEXT     ;BR, wenn keiner mehr da ist
LDY    #CPFNAM    ;ansonsten hole File-
LDA    (CPALOC),Y ;Namen aus dem Buffer
```

10.2.6 RUNLOC

Jedesmal, wenn ein externes Kommando aufgerufen wird, erhält RUNLOC ($3D im CP) den Wert der ersten Adresse im .COM-File des betreffenden Kommandos. Einige externe Befehle) einschließlich vom Benutzer selbst geschriebene) können deshalb mit RUN auch neu gestartet werden. Es kann die Notwendigkeit auftreten, daß man den Inhalt von RUNLOC verändern will, so daß er auf den Warmstartpunkt eines eigenen Programmes deutet, wenn es zum ersten Male geladen wird, und eine ungewünschte Reinitialisierung beim erneuten Entern entfällt. Unter BASIC A+ und MAC/65 wird dies ausgenutzt, damit ein evtl. im Speicher befindliches Programm nach einer Rückkehr aus dem CP, auch noch wirklich da ist. Möchten Sie ein erneutes Entern vermeiden, so müßten Sie das höherwertige Byte ($3E im CP) in RUNLOC in Null ändern:

```
        LDY   #RUNLOC+1     ;erneutes Entern unterdrücken
        LDA   #0
        STA   (CPALOC),Y
```

10.3 Die Device-Handler (Verwalter)

Wie bereits angedeutet, ist die CIO ein wirklich kleines Programm (ca.700 Bytes), ist aber trotzdem in der Lage, das weite Feld der Ein-/Ausgabe-Anforderungen, die in Kapitel 10.1 und 10.2 beschrieben wurden, mit einem erstaunlich einfachen und konsistenten Assembler-Interface abzuwickeln. Noch einfacher aufgebaut ist die Schnittstelle zwischen dem OS und den Device-Verwaltern.

Aus diesem Grunde ist allerdings die CIO auch oft langsamer als man es sich wünschen würde, was man aber auch nur bei "PUT BINARY RECORD" und "GET BINARY RECORD" bemerkt und die Handler müßen ziemlich kompliziert aufgebaut sein. So schlimm ist es aber gar nicht, was in den folgenden Unterkapiteln beschrieben wird.

10.3.1 Die Device-Handler-Tabelle.

In der Speicherstelle "HATABS" im RAM, (die aus dem ROM geladen wird) liegt eine Liste des CIO mit Standard-Devicenamen (P:,D:,E:,S: und K:) und deren Adressen. Um ein neues Device ansprechen zu können, muß es einfach an die bestehende Liste angehängt werden; d.h. der Device-Name (ein Buchstabe) und die Adresse der zugehörigen Handler-Tabelle wird angegeben (Hierzu gleich mehr).
Theoretisch können alle genannten Device-Handler unter DOS XL mehr als ein physikalisches Device verwalten, so wie der Disk-Handler "D1:","D2:" usw. versteht, könnte ein Tastaturverwalter "K1:","K2:" etc. verwalten. DOS XL hängt eine Unterdevice-Nummer (die 1) an eine Devicebuchstabenangabe, wenn diese nicht näher spezifiziert ist (d.h. D: wird zu D1:).

Im folgenden ist die HAndler TABelle der ATARI-Computer zu finden:

```
    *=      $031a
HATABS
    .BYTE   'P'     ; Printer
    .WORD   PDEVICE ; und die Adresse seines Drivers
    .BYTE   'C'     ; Cassettenrecorder
    .WORD   CDEVICE
    .BYTE   'E'     ; Bildschirm
    .WORD   EDEVICE
    .BYTE   'S'     ; GRAPHIC SCREEN
    .WORD   SDEVICE
    .BYTE   'K'     ; Tastatur
    .WORD   KDEVICE
    .BYTE   0       ; die Null markiert das Ende der Tabelle
    .WORD   0       ; jedoch ist hier noch Platz für
    .BYTE   0       ; andere Devices etc...
```

10.3.2 Richtlinien zur Erstellung von Device-Handlern.

Von jedem Device, dessen Adresse in der Handlertabelle (siehe oben) steht, wird die Beachtung einiger Regeln erwartet; genau gesagt, wird vorausgesetzt, daß ein Driver sechs (6) Aktionsunter- und eine Initialisierungsroutine besitzt (In der Praxis ruft das momentan in die ATARI's eingebaute OS nur die Initialisierungsroutine für seine eigenen vordefinierten Devices auf. Da sich dies in Zukunft ändern könnte und man sowieso den Aufruf einer eigenen Initialisierungsroutine bewirken kann, muß man voraussetzen, daß jeder Driver berücksichtigt wird, auch wenn er gar nichts enthält.). Die in der Handlertabelle abgelegte Adresse muß auf eine andere Tabelle zeigen, dessen Form unten gezeigt wird:

```
HANDLER
   .WORD  <Adresse der OPEN-Routine>-1
   .WORD  <Adresse der CLOSE-Routine>-1
   .WORD  <Adresse der GETBYTE-Routine>-1
   .WORD  <Adresse der PUTBYTE-Routine>-1
   .WORD  <Adresse der STATUS-Routine>-1
   .WORD  <Adresse der XIO-Routine>-1
   JMP    <Adresse der Initialisierungsroutine>
```

Tabelle 10.3

Beachten Sie bitte die sechs Adressen, die angegeben werden müssen und, daß man jeweils 1 von der Adrese abziehen muß (das -1 erleichtert der CIO die Arbeit wesentlich). Eine kurze Beschreibung jeder Routine wird auf den folgenden Seiten gegeben.

Device OPEN

Die OPEN-Routine führt jede, vom Device benötigte Initialisierung aus. Für viele Devices, so wie z.B. den Drucker, kann dies einfach aus einer kurzen überprüfung des Device-Status auf sein Vorhandensein bestehen. Da das X-Register beim Eintritt in eine solche Routine, die für diesen Aufruf benutzte IOCB-Nummer, beinhaltet, kann der Driver ICAUX1 (über LDA ICAUX1,X) und/oder ICAUX2 dies nach der Art des angeforderten OPEN überprüfen.

<u>Achtung:</u> Die CIO sieht Bit 2 und 3 ($04 und $08) sowohl für Schreib- als auch für Lese-Kontrolle vor. Diese Bits können zwar gelesen, sollten aber nicht verändert werden.

Device CLOSE

Die CLOSE-Routine ist oft noch einfacher, sie sollte das DEVICE, wenn nötig und möglich, schließen.

Device PUT- und GET-Routinen

Die PUT-und GETBYTE-Routinen sind einfach das, was ihr Name schon verrät: der Device-Handler muß eine Routine bereithalten, die die Ausgabe eines einzelnen Bytes an das Device und eine die, die Eingabe eines Bytes vom Device, ermöglicht. Für einige Devices ergibt die eine oder andere Routine keinen Sinn (Haben Sie schon einmal ein INPUT vom Drucker versucht?). In einem solchen Fall kehrt die Routine einfach mit RTS ins DOS XL zurück und eine entsprechende Fehlermeldung wird ausgegeben.

Device STATUS-Routine

Die STATUS-Routine ist dafür vorgesehen einen dynamischen STATUS-Check durchzuführen. Ist eine dynamische überprüfung nicht erwünscht oder durchführbar, meldet die Routine einfach den Status-Wert des User-IOCB zurück.
Vorsicht: Unter ~~DOS~~ XL tritt keine Fehlermeldung auf, wenn man versucht den STATUS eines ungeöffneten Devices abzufragen.

Device erweiterter Ein-/Ausgabe-Routinen.

Auch die XIO-Routine tut das, was ihr Name schon enthält: Sie erlaubt dem Benutzer jede und alle speziellen Routinen, die ein entsprechender Device-Handler bereithält, aufzurufen. Das Betriebssystem verändert an einem solchen Aufruf gar nichts und gibt diesen einfach an den dafür zuständigen Driver weiter.

Generelle Anmerkungen zu Device I/O-Routinen.

Allgemein sind die AUXiliary (Hilfs-)Bytes jedes IOCB's für jeden Driver verfügbar. In der Praxis ist es jedoch am besten, wenn man ICAUX1 und ICAUX2 meidet, da diese von verschiedenen BASIC und OS-Kommandos verändert werden. Beachten Sie bitte, daß ICAUX3 bis ICAUX5 zum Informationsempfang sowie zur Infomationsweitergabe vom/zum BASIC über die NOTE und POINT-Kommandos, die wirklich spezielle XIO-Kommandos sind, verwendet werden können. Schließlich sollten die Driver keine anderen Bytes in den IOCB's beeinflußen, speziell nicht die ersten zwei.

Die Handler müssen auch nicht mit PUT BINARY RECORD und GET BINARY RECORD belästigt werden, da das OS die Verwaltung dieser User-Level-Kommandos übernimmt.

10.3.3 Richtlinien zur Erweiterung des Betriebssystems.

1. Untersuchen Sie den MEMLO-Zeiger (siehe SYSEQU.ASM für die aktuelle Speicherstelle)
2. Laden Sie Ihre Routine (einschließlich benötigter Buffer) an den Wert aus MEMLO.
3. Addieren Sie die Größe Ihrer Routine zu MEMLO.
4. Speichern Sie den sich ergebenden Wert wieder in MEMLO ab.
5. Schließen Sie Ihren Driver an das OS an, indem Sie dessen Namen und Adresse in die Handler-Tabelle eintragen.
6. Umgehen Sie das OS, indem bei einem Druck auf RESET die Schritte 3 bis 5 erneut ausgeführt werden, da SYSTEM RESET wirklich die Handlertabelle und den Wert in MEMLO reseted.

Hier ist Schritt 2 am schwersten durchzuführen, da man für das Einladen der Routine an die Stelle an der sich MEMLO gerade befindet eine verschiebbare (oder sich selbst verschiebende) Routine benötigt. Da es für ATARI-Computer ~~keinen~~ Assembler gibt, der einen solchen verschiebbaren Code herstellen kann, ist dies keine leichte Aufgabe. Allerdings ist dies auch nicht unbedingt nötig, wenn Sie sowieso nur Programme für den privaten und nicht den öffentlichen Gebrauch schreiben.

Schritt 6 läßt sich realisieren, indem man das Betriebssystem glauben läßt, daß der eigene Driver der Disk-Driver zu Initialisierungszwecken ist (man leiht sich den DOSINI-Vektor aus) und dann den Disk-Initializer selbst aufruft, bevor die Schritte 3 bis 5 erneut ausgeführt werden.

10.3.4 Ein Beispielprogramm

Dieser Driver, der auf der Masterdisk unter dem Namen MEM.LIS abgespeichert ist, erstellt einen neuen Driver und fügt diesen zum Betriebssystem hinzu. Das "Device" ist einfach ein Speicherbereich innnerhalb des Computers; dieser Bereich könnte z.B. als Pseudo-Diskfile zur Weitergabe von Daten an sequentiell aufgerufene Programme dienen.

Einige Dinge sind allerdings zu beachten. Dieser Driver führt Schritt 6, der im vorhergehenden Kapitel angegeben wurde, nicht aus (kann aber mit einem BASIC USR-Aufruf hineinitialisiert werden). Eine Selbstverschiebung wird auch nicht ausgeführt, dafür platziert es sich aber über die normale Nutzung des untersten Speicherbereichs (außer den seriellen Port-Drivern, die nach diesem Driver geladen werden müßten). Wenn Sie es selbst assemblieren, könnten Sie dies an dem normalen MEMLO in Ihrem System durchführen oder Sie könnten es natürlich gleich selbstmodifizierend machen).

Einen kleinen Haken hat es noch: es benutzt das RAM vom MEMTOP (oberste Speichergrenze) aus nach unten und überprüft hier nicht, ob es bereits den BASIC MEMTOP ($90) überschritten und damit evtl. Programm oder Daten überschrieben hat. Um ganz sicher zu gehen, schreiben Sie nicht mehr Daten, als ein FRE(0) anzeigt oder sogar noch etwas weniger.

Während der Ausführung wird der M:-Driver über einen OPEN-Befehl für Schreibzugriff (Modus 8) hineinitialisiert. Ein CLOSE gefolgt von einem subsequenten Lese-Zugriff erlaubt es, die Daten in der Reihenfolge, in der diese geschrieben wurden, zu lesen.

Achtung: Verändern Sie nicht die Graphikmodi zwischen Schreiben und Lesen, wenn diese Änderung mehr Speicher benötigt (um sicherzugehen, verändern Sie da besser gar nichts). Das M:-Device verhält sich fast so wie ein Cassettenfile, so daß das Userprogramm wenn nötig die eingelesenen Daten richtig deuten ~~können~~ muß, da hier keine Fehlermeldungen ausgegeben werden. Beachten Sie bitte, daß die Daten auch mehrmals (via CLOSE und re-OPEN) gelesen werden können.

Beispielprogramm:

Ende von Programm 1:

```
9900 OPEN #2,8,0,"M:"
9910 PRINT #2;LEN(A$)
9920 PRINT #2;A$
9930 CLOSE #2
9940 RUN "D:PROGRAM2"
```

Anfang von Programm 2:

```
100 OPEN #4,4,0,"M:"
110 INPUT #4,SIZY
120 DIM STRING$(SIZE)
130 INPUT #4,STRING$
140 CLOSE #4
```

BASIC XL-Benutzer werden merken, daß man hier als nützliche Werkzeuge RPUT/RGET und BPUT/BGET einsetzen kann. Benutzer anderer Sprachen könnten dieses kleine Beispiel als brauchbares Kommunikationsmittel zwischen Programmen verwenden.

Kapitel 11: Filestrukturen

DOS XL Version 2 wurde unter dem Gesichtspunkt der maximalen Kompatibilität mit den ATARI DOS's hergestellt. Tatsächlich ist das benutzte FMS mit dem von ATARI identisch (da OSS das FMS für ATARI geschrieben hat). Aus bestimmten Gründen wurde das ATARI-FMS nach einem Verbund-Sektoren Disketten-Management-Schema erstellt; d.h. daß die letzten drei Bytes eines Sektors innerhalb eines Diskfiles eine Verzweigung zum nächsten Sektor innerhalb dieses Files beinhalten. Das positive Ergebnis ist, daß man einen relativ kleinen, speicherresistenten Diskmanager hat, der in der Lage ist den Speicherbereich einer Diskette voll auszunutzen (im Gegensatz zu einem fortlaufenden Diskmanager). Der größte Nachteil eines solchen Schemas scheint zu sein, daß man keinen direkten, wahlfreien Zugriff auf die einzelnen Bytes eines solchen Files hat, so wie man das bei einer fortlaufenden oder gekennzeichneten File-Allocation-Technik hätte. Außerdem bewirkt ein Fehler in der Filekette in der Mitte des Programmes den Verlust des Gesamtfiles.

Der Zweck des FMS ist es, die 720 (oder 1040 bei Enhanced Density auf einer 1050), auf einem 810-Laufwerk verfügbaren Datensektoren in ein System mit Namen versehener Files zu organisieren. Das FMS hat drei primäre Datenstrukturen zur Organisation der Diskette:

1. Volume Table of Contents (VTOC= Inhaltsverzeichnistabelle): ein einzelner Sektor auf der Disk, der anzeigt welche Sektoren zur Verwendung in Datenfiles frei sind.

2. Directory (Inhaltsverzeichnis): eine Gruppe von acht aufeinanderfolgenden Sektoren in der Mitte der Diskette, die sowohl die Namen, der sich auf der Disk befindlichen Files, als auch deren Adressen beinhaltet. Jeder Directory-Eintrag enthält einen Filenamen, einen Zeiger auf dessen ersten Datensektor und einige untergeordnete Informationen.

3. Datensektoren: sind die Sektoren, die die eigentlichen Daten und einige Kontrollinformationen, die einen Sektor mit dem nächsten verbinden, enthalten.

Beachte: Da die ~~Sekto~~ren bei Disketten in doppelter Schreibdichte 256 Bytes pro Sektor beinhalten, bei einfacher Dichte (810) jedoch nur 128 Bytes, kann die absolute Byteanzahl von Disk zu Disk variieren. Aus diesem Grund wird die Anzahl für single-density Sektoren im folgenden gefolgt von der in double-density in Klammern, angegeben.

11.1 Datensektoren.

Ein Datensektor wird verwendet, um die Datenbytes eines Files zu speichern. Jeder 128 (256) Byte-Datensektor ist so organisiert, daß in ihm 125 (253) Datenbytes und 3 Kontrollbytes stehen. Die Datenbytes innerhalb eines Sektors starten mit dem ersten Byte (Byte 0), die nächsten folgen in aufsteigender Reihenfolge bis einschließlich Byte 124 (252). Die Kontrollinformationen folgen ab Byte Nummer 125 (253).

Der Sektoren Bytezähler liegt in Byte 127 (255) und gibt die Anzahl der in diesem Sektor belegten Datenbytes an. Der Wert kann von 0 (keine Daten) bis 125 (253) (voller Sektor) variieren. Jeder Datensektor innerhalb eines Files kann ein sog. kurzer Sektor sein und weniger als 125 (253) Datenbytes enthalten.

Die übrigen sechs Bits von Byte 125 (253) beinhalten die Filenummer des Files, zu dem der Sektor gehört. Diese Nummer entspricht der Nummer, die das File bei seiner Speicherung im Directory erhalten hat. Der erste Eintrag in den Directory-Sektor $169 erhält die Filenummer 0, der zweite Eintrag die Nummer 1 usw. Die Filenummer kann von 0 bis 63 reichen, so daß maximal 64 Filenamen im Directory Platz finden. Diese Nummer wird verwendet, damit die Sektoren der verschiedenen Files nicht verwechselt werden.

Die rechten beiden Bits von Byte 125 (253) und alle 8 bits von Byte 126 (254) werden verwendet, um auf den nächsten Datensektor innerhalb des Files zu deuten. Diese zehn-bit Nummer bestimmt die Nummer des nächsten Sektors; der Wert variiert von 0 bis 719 ($2CF). Ist der Wert 0, so folgen keine weiteren Sektoren in der Filesektorkette. Der letzte Sektor in dieser Kette ist der ENd-of-file-Sektor und ist meistens ein kurzer Sektor.

11.2 Disk Directory

Das Directory beginnt bei Sektor $169 und wird in den darauffolgenden 8 Sektoren fortgeführt, beendet von Sektor $170. Diese Sektoren wurden für das Directory ausgesucht, da sie in der Mitte der Disk liegen und daher die durchschnittlich kürzeste Zugriffszeit von jeder Stelle aus haben. In jedem Directory-Sektor ist Platz für 8 Fileeintragungen, so daß es möglich ist max. 64 Files auf einer Diskette zu verwalten.

Ein Direcotery-Eintrag hat eine Größe von 16 Bytes, wie in Tabelle 11.1 gezeigt. Das Directory-Entry-Flag-Feld gibt spezielle Statusinformationen über den aktuellen Eintrag. Im Directory-Count-Feld werden die von einem File momentan belegten Sektoren gespeichert. Die letzten 11 Bytes des Eintrags sind der aktuelle Filename. Der primäre Filename wird entsprechend im primären Filenamen-Feld abgelegt. Unbenutzte Filenamenstellen sind Leerzeichen ($20). Das Startsektornummer-Feld deutet auf den ersten Sektor des Files.

Startbyte # Nummer des Feldes	Länge des Feldes (in Bytes)	Zweck des Feldes
0	1	Flag Byte. Bedeutung der Bytes: $00 Eintrag noch nie benutzt $80 Eintrag wurde gelöscht $40 Eintrag wird benutzt $20 Eintrag ist geschützt $02 ein Version 2 File $01 File wird gerade geschr.
1	2	Anzahl (LSB,MSB) der Sektoren im File
3	2	Startsektor (LSB,MSB) der Kette
5	8	Filename, primär
13	3	Filename, Extender

Tabelle 11.1

Directory Eintragungstruktur

```
      (sektor $169)
     erster Directory
         sektor
     ----------------
     ! FileA  !Zeiger! --->>   ----------
    -! FileB  !Zeiger! --+     !Sektor 1!
  -! !  etc.  !      !   !     !  von   !
 ! ! !        !      !   !     ! File A !
 ! ! !        !      !   !     ! -------!
 ! ! !        !      !   !     ! !Zeiger!-->>----------
 ! ! !        !      !   !     !--------!   !weitere !
 ! ! !        !      !   !                  !File A  !
 ! ! ----------------    !                  !Sektoren!
 ! !(Sektor $16A)  !     !                  !--------!
 ! ----------------      !
 !   etc.          !     !
 -----------------       +-->> ----------
                               !Sektor 1!
                               !  von   !
                               ! File B !
                               !--------!
                               !  !Zeiger! -->> etc.
                               ----------
```

Schaubild 11.2

Version 2 Directory Struktur

Beachte: es sind nur 8 Eintragungen pro Sektor zugelassen, auch bei doppelter Dichte.

11.3 Volume Table of Contents (VTOC).

Der VTOC-Sektor ($168) wird verwendet, um darauf aufzupassen, welche Sektoren für Datenfiles noch frei sind. Tabelle 11.3 zeigt die Organisation des VTOC-Sektors, wobei das wichtigste Teil die Sektoren Bit-Map ist.

Die Sektor.Bit-Map ist ein aufeinanderfolgender String von 90 Bytes, von denen jeder 8 Bit enthält; daraus ergibt sich eine Summe von 720 (90 x 8) Bits in der Bit-Map, einer für jeden möglichen Sektor auf einer 810er Disk. Die 90 Byte der Bitmap beginnen mit dem VTOC-Byte zehn ($0A). Das Bit ganz links ($80 bit) von Byte $0A repräsentiert den Sektor 0. Das nächste Bit nach rechts ($40 bit) Sektor 1 usw. Das Bit ganz rechts ($01 bit) von Byte $63 dementsprechend den Sektor 719.

Startbyte Nummer des Feldes	Länge des Feldes (in Bytes)	Zweck des Feldes
0	1	Reserviert (für Typencode)
1	2	Gesamtanzahl der Sektoren
3	2	Anzal der unenutzten S.
5	5	Reserviert.
10	90	Bit-Map der benutzten S. Jedes bit repräsentiert einen bestimmten Sektor: eine 1 zeigt einen verfügbaren , eine 0 einen bereits besetzten Sektor an.
100	28	Reserviert (Könnte bei Version 2 für mehr als 720 Sektoren verwendet werden.

Tabelle 11.3

Struktur des VTOC-Sektors

Anhang A: Anpassung von DOS XL.

Obwohl DOS XL für den normalen Benutzer geschrieben und implementiert wurde, kann keine Software jemals allen Anwendern gerecht werden. Aus diesem Grunde wurde in verschiedenen Richtungen eine gewisse Flexibilität gewahrt, so daß das System dem Benutzer erlaubt, eigene Modifikationen vorzunehmen. Im folgenden werden die meistgebrauchtesten und sinnvollsten Modifizierungen beschrieben.

A.1 Buffer-Verlegung.

DOS XL erlaubt dem Benutzer die Startadresse des System-Filebuffers und die Anzahl der verwendeten Buffer selbst festzulegen. Allerdings ist es nicht garantiert, daß die Speicherstellen, die dies bestimmen auch in Zukunft an ihrem derzeitigen Platz bleiben, so daß dringend geraten wird, daß derjenige, der in dieser Richtung Veränderungen vornehmen will, am besten zuerst die Parameter im File SYSEQU.ASM auf ihre aktuellen Werte hin überprüft. Zur Zeit der Erstellung dieses Handbuches sind folgende Werte aktuell:

Label	Speicherst.	Gebrauch
SASA	$070C	Start der Buffer
SABYTE	$0709	Anzahl der Buffer

Angenommen der Benutzer möchte SABYTE verändern, stellt sich zuerst folgende Frage: "Wieviele Buffer benötige ich ?"; Beachten Sie bitte hierzu folgende Grundregeln:

Benötigter Speicherplatz: Verwenden Sie für einfache Schreibdichte 2 Buffer pro aktives Laufwerk und 1 Buffer für simultan geöffnete Files. Bei doppelter Dichte: 2 Buffer pro aktives Drive und 2 für simultan geöffnete Files. Jeder Buffer ist 128 Zeichen lang.

Vergewissern Sie sich, daß an der Stelle, die Sie in SASA spezifizieren genügend Speicherplatz ist, um die Buffer aufzunehmen.

Anzahl der Buffer festlegen: Schließen Sie die Drive-Buffer nicht in die Addition der Buffer, die Sie DOS XL zur Verfügung stellen, mit ein, sondern spezifizieren Sie lediglich die FILE-Buffer in Speicherstelle SABYTE.

A.2 Spezifizieren existierender Laufwerke.

Unter Version 2 kontrolliert die Byte-Speicherstelle DRVBYT (in $70A, aber sehen Sie lieber noch einmal in SYSEQU.ASM nach), welche Laufwerke aktiv sind. Jedes Bit von DRVBYT repräsentiert ein Laufwerk. Das Bit mit dem niedrigsten Wert die Station Nummer 1 und das mit dem höchsten LW 8.

Ist ein Bit in DRVBYT gesetzt (d.h. hat es den Wert 1) ist das Laufwerk aktiv, hat das entsprechende Bit den Wert 0, dann ist es inaktiv. Ein Wert von $05 würde z.B. bedeuten, daß D1: und D3: aktiv sind.

Achtung: Es reicht nicht aus einfach ein Bit in DRVBYT zu ändern oder Informationen zur Diskdrive-Tabelle hinzuzufügen, um die Systemkonfiguration zu modifizieren. Nachdem die Bits geändert wurden, muß man DOS XL veranlaßen sich neu zu initialisieren, was man einfach durch einen Druck auf >SYSTEM RESET< von der Tastatur aus oder durch Aufruf der DOS-Initialisierungsroutine über DOSINI aus einem laufenden Programm bewirken kann.

A.3 Das Abspeichern der modifizierten Version

Dies ist extrem einfach: Verwenden Sie unter Version 2 einfach das INIT-Kommando und wählen Sie aus dem daraufhin erscheinenden Menü: "Write DOS.SYS-File only" aus (oder Sie nehmen eine ganz neue Disk und formatieren diese gleich mit; löschen Sie aber keine wichtigen Daten). Diese Option kann aber nicht bei der Verwendung eines erweiterten Speicher-DOS-Systems gebraucht werden (Sehen Sie hierzu bitte in den Kapitel 3.7, 6.8 und 7.7 für weitere Informationen nach).

Anhang B: DOS XL und das 850 Interface Modul.

B.1 Das Einladen des RS232-Handlers.

Unter den ATARI-DOS 2.x ist die einzige Möglichkeit den RS232-Device-Handler (Rn:) vom 850-Modul einzuladen, durch die Verwendung eines AUTORUN.SYS-Files gegeben (Siehe Kapitel 8.2). Diese Option ist auch unter DOS XL anwendbar, jedoch gibt es noch eine andere Möglichkeit. Nach dem Booten von DOS XL sind einfach folgende Kommandos einzugeben:

1. vom DOS XL Menü aus tippen Sie bitte ein:
   X und dannach auf das Prompt:
   RS232 >RETURN<
2. oder vom CP aus:
   RS232 >RETURN<

Daraufhin wird der RS232-Device-Handler in das System eingeladen und man kann auf die 4 RS232-Ports (R1: bis R4:) im 850-Interfacemodul zugreifen.

B.2 Fehler im RS232-Driver.

Leider ist der Device-Handler, der vom 850-Modul eingeladen wird nicht perfekt. Der schlimmste Fehler tritt auf, wenn man nach dem Laden des RS232.COM-Files >SYSTEM RESET< drückt. Unter bestimmten Umständen hängt sich der Computer auf und man hat keine Kontrolle mehr über die Tastatur. Aus diesem Grunde raten auch die meisten ATARI-Handbücher, in denen die 850 vorkommt, niemals >SYSTEM RESET< zu betätigen, nachdem der Handler installiert wurde. Unter DOS XL gibt es aber noch eine andere Lösung: Auf der Masterdisk befindet sich auch das File RS232FIX.COM, das fast identisch mit RS232.COM ist, bis auf die korrigierten Fehler.

Nun wird man sich fragen: "Warum ist dann nicht nur die korrigierte Version auf der Diskette ?". ATARI hat im Laufe der Zeit verschiedene Versionen des 850 Interface-Modules auf den Markt gebracht und es ist nicht sichergestellt, daß alle diese Versionen mit der korrigierten Version des RS232-Handlers zusammenarbeiten. Um keine erneuten Probleme auftreten zu lassen, wurden beide Versionen implementiert. Um Ihr 850 auf die Zusammenarbeit zu testen, gehen Sie bitte wie folgt vor:

1. vom DOS XL Menü aus:
   X und dann:
   RS232FIX >RETURN<
2. oder vom CP aus:
   RS232FIX >RETURN<

Wenn, der auf diese Weise geladene Handler funktioniert, sollten Sie lieber diese Version verwenden.

Anhang C: System Speichertabellen.

C.1 Atari Zero Page Tabelle.

Speicherst.		Verwendung
0-9		System Zero Page
A-D	CPALOC	DOSVEC bei ATARI DOS
C-D	DOSINI	Vektor zur FMS-Initialisierung
E-42		System Zero Page
43-49		FMS Zero Page
4A-7F		System Zero Page
80-FF		Benutzer und Sprachen- Zero Page
80-CD		BASIC XL und ATARI BASIC Zero Page
D2-FF		Fließkomma Zero Page

C.2 ATARI System Speichertabelle - DOS XL Version 2.

Speicherst.		Verwendung
100-1FF		6502 Stack Bereich
200-319		System RAM
300-30B		Device Control Block (DCB)
31A-33F		Device Handlertabelle
340-3BF		IOCB's - 8 a 15 Bytes
3C0-57F		System RAM
580-5FF		E: Textbuffer
600-6FF		User RAM
700-variiert		DOS XL - Filemanager und CP oder nur Buffer etc., bei der Benutzung des erweiterten Speicher DOS Systems.
709	SABYTE	Anzahl der 128-Byte Buffer
70A	DRVBYT	Bit Map: aktive Drives
70C	SASA	Startadresse der Buffer
(2E7)-BFFF		Benutzersprache und Graphikspeicher; (2E7) bedeutet: Inhalt der Speicherstelle $02E7 (LOMEM)
A000-BFFF		SuperCartr. und ATARI BASIC Speicher oder auch DOS XL Filemanager und CP Bei erweitertem System.
C000-CFFF		IN 400/800 unbenutzt; OS ROM bei XL-Serie, bankswitched mit dem RAM.
D000-D7FF		I/O-Speicherstellen
D500-D5FF		SuperC.: zum Bankselect benutzt
D800-DFFF		Fließkomma ROM
E000-E3FF		Character Set ROM
E400-FFFF		OS Driver, CIO etc. im ROM
E400-FFF9		Bankselectbares RAM,bei DOS XL für Filamanager und CP, etc. bei DOSXL.XL-Version mit erweitertem Speicher

---

Anahng D: ATARI Schreiber und andere Cartridges.

Bestimmte auf Modulen basierende Produkte, einschließlich ATARI-Schreiber von ATARI, Inc. funktionieren nicht einwandfrei, wenn sich auf der bootenden Disk ein STARTUP.EXC-File befindet. Sollten Sie ein solches Modul verwenden, erstellen Sie sich bitte eine neue Bootdisk wie folgt:

1) Duplizieren Sie Ihre Masterdisk auf eine leere Disk.
2) Löschen Sie das File STARTUP.EXC auf der neuen Disk.
3) Löschen Sie das File MENU.COM, wenn Sie mehr Speicherplatz zur Verfügung haben möchten.

Nun können Sie diese Diskette zum Booten in ATARI-Schreiber oder andere Module verwenden, haben aber keinen Zugang zum DOS XL Menü.

Anhang E: Fehler

E.1 Fehlertypen

Alle DOS XL Operationen melden einen Statuswert an das IOSTAT-Feld. Hier ist festgelegt, daß ein Statuswert von $80 oder höher einen Fehler anzeigt. Es gibt 4 fundamentale Fehlersorten, die unter DOS XL auftreten können.

Hardware Fehler
z.B. beim Versuch eine defekte Diskette zu lesen, auf eine nur zum lesen geöffnete Disk schreiben etc.

Datentransfer-Fehler
Fehler, die während dem Datenaustausch zwischen Computer und einem Peripheriegerät auftreten. z.B. Device Timeouts, Device NAK, Framing Errors etc.

Device Driver Fehler
Begründet im Driver für das entsprechende Device, wie bei File not found (File nicht gefunden), File locked (File schreibgeschützt), Invalid Drive number (Ungültige Laufwerksnummer). etc.

Betriebssystem-Fehler
Gewöhnlich fundamentale Anwendungsprobleme, wie Bad Channel Number (Falsche Kanalnummer), Bad Command (Falsches Kommando), etc.

E.2 Liste der Fehlermeldungen.

Die nun folgende Liste der Fehlermeldungen ist nicht unbedingt vollständig, zeigt aber alle Fehlermeldungen, die normalerweise von DOS XL oder von einem der ATARI Devices gemeldet werden.

Fehlercode

HEX	DECIMAL	Bedeutung:
$01	1	Kein Fehler
$02	2	Truncated ASCII-Zeile;das OS hat keinen CR innerhalb von BUFLEN für den ASCII-Zeilen I/O gefunden.
$03	3	End of File kommt gleich. Das letzte vom Device-Driver gelieferte Byte war ein End-of-File Byte. Der Driver muß auf jeden Fall einen Status liefern, so daß man sich vorher vergewissern muß, ob das angeschl. Device überhaupt zu diesem Status fähig ist.
$80	128	Operation abgebrochen,wird vom Device-Handler gesetzt; kann beim ATARI auch Abbruch durch BREAK-Taste bedeuten.
$81	129	File bereits geöffnet. Das Programm hat versucht einen bereits geöffneten Kanal (IOCB) zu öffnen.
$82	130	Device existiert nicht. Das Device wurde in der OS Device-Tabelle nicht gefunden. Tritt oft auf, wenn man vergißt die LW-Nummer in Verbindung mit einem File anzugeben.
$83	131	File ist nur zum Schreiben geöffnet. Das Programm hat versucht aus einem File zu lesen, das nur zum Schreiben geöffnet wurde, d.h. AUX1 war auf 8 oder 9 gesetzt, während das File geOPENed wurde.
$84	132	Ungültiges Kommando. Die CIO weist das eingegebene Kommando zurück. (z.B. das Programm hat versucht ein XIO zu einem Device zu schicken, der dort gar nicht ausgeführt werden kann).
$85	133	Device/File nicht geöffnet. Der IOCB wurde für diese Operation nicht geöffnet. Die meisten I/O-Anforderungen setzen voraus, daß vorher ein Kanal für sie geöffnet wurde.
$86	134	Der angegeben IOCB ist ungültig. Nur die IOCB Nummern $00, $10, $20, $30, $40, $50, $60 und $70 sind zulässig. Von einigen Sprachen werden diese als Kanal 0 bis 7 angesehen.

$87	135	File nur zum Lesen geöffnet. Es wurde versucht in ein File zu schreiben, das nur zum Lesen geöffnet war, d.h. AUX1 war auf 4 oder 6 gesetzt, während das File geOPENed wurde.
$88	136	End of File: das Ende des Files wurde erreicht, d.h.keine Daten mehr.
$89	137	Truncated Record-Fehler. Tritt gewöhnlich auf, wenn die Zeile, die gelesen wird länger als die max.Record-Größe, die im CIO-Aufruf spezifiziert ist (zeilenorientierter I/O), ist. Kann unter Version 2 des OS/A+ nicht auftreten.
$8A	138	Device Timeout-Fehler. Wird meist vom seriellen Bus-I/O-Handler (SIO) gesetzt, da ein Device nicht im vom OS vorgeschriebenen Zeitraum geantwortet hat.
$8B	139	Device NAK-Fehler. ATARI: serieller I/O-Fehler
$8C	140	Serieller Rahmen-Fehler.
$8D	141	Cursor außerhalb der im aktuellen Graphikmodus erlaubten Bildschirmbereich; könnte auch von einem graphikfremden aber ähnlichem Device verwendet werden.
$8E	142	Serieller Bus ist überladen. Der Computer kommt nicht mit dem seriellen Bus-Input mit (SIO-Fehler).
$8F	143	Prüfsummen Fehler. Die Kommunikation auf dem seriellen Bus ist durcheinander (SIO-Fehler).
$90	144	1) vom Device verursachter Fehler: ein erlaubtes Kommando wurde auf dem seriellen Bus nicht richtig ausgeführt. 2) Schreibschutzfehler: Die Aussparung an der Diskette ist mit einem Schreibschutzaufkleber bedeckt.
$91	145	Illegaler Screenmodus-Fehler: Falsche Graphikmodus Nummer. Bei anderen Devices: Die AUX1- und/oder AUX2-Bytes im IOCB sind illegal.
$92	146	Die angeforderte Funktion ist nicht im Devicehandler implementiert, z.B. wurde versucht den POINT-Befehl in einem Graphik-Device anzuwenden.

$93	147	Nicht genügend RAM für den angewählten Graphikmodus; kann auch von ähnlichen Devices verwendet werden.

Anmerkung: $A0 bis $AF sind Filemanager-Fehler.

$A0	160	Entweder wurde eine Drive-Nummer außerhalb von 1-8 angegeben oder das betreffende Laufwerk ist nicht eingeschaltet.
$A1	161	Zu viele OPEN-Files. Keine freien Sektorbuffer für ein weiteres File mehr vorhanden.
$A2	162	Diskette voll.
$A3	163	Fataler System Fehler. Entweder hat das DOS einen Fehler oder die Diskette ist defekt.
$A4	164	Durcheinander innerhalb eines Files. Falsche Filestruktur oder falsche POINT-Werte.
$A5	165	Ungültiger Filename. Falsche Zeichen bei einer Filenamenangabe. Version 4 nimmt es hier nicht so genau wie Version 2.
$A6	166	Die Summe der Bytes in einem POINT-Aufruf war größer als 125 (für einfache Dichte in Version 2) oder größer als 253 (bei doppelter Dichte in Version 2).
$A7	167	Das angegebene File ist schreibgeschützt und kann weder beschrieben noch gelöscht werden.
$A8	168	Das Software-Interface für das angegebene Device hat einen ungültigen Befehl erhalten (z.B. wurde ein nicht existenter Track oder Sektor angesprochen).
$A9	169	Directory voll: Zu viele verwendete Filenamen.
$AA	170	Das ausgewählte File ist nicht auf der Disk.
$AB	171	Es wurde versucht auf ein Byte innerhalb eines Files zu deuten, das nicht für Update geöffnet wurde (nur bei Version 2).
$AC	172	Es wurde versucht ein DOS I-File von DOS II zu OPENen (nur bei Version 2).
$AD	173	Während der Formatierung wurden defekte Sektoren entdeckt.